

Д. Б. ЮДИН,
Е. Г. ГОЛЬШТЕЙН

Линейное программирование

ТЕОРИЯ,
МЕТОДЫ И ПРИЛОЖЕНИЯ



ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1969

Линейное программирование (теория, методы и приложения), Д. Б. Юдин, Е. Г. Гольштейн.

Книга содержит подробное систематическое изложение теории, методов и приложений общей задачи линейного программирования. Первая глава носит вводный характер. Глава 2 посвящена описанию ряда практических задач (в основном экономического происхождения), приводящихся к общей схеме линейного программирования. Математический аппарат линейного программирования, включая теории многомерных множеств и двойственности, собран в главе 3.

Последующие главы книги дают весьма полное представление о существующих конечных методах линейного программирования.

При решении на ЭВМ задач линейного программирования больших размеров со слабозаполненными матрицами условий (именно такие задачи, как правило, и возникают в экономике) весьма эффективными оказываются бесконечные итеративные процедуры, многие из которых носят игровой характер. Взаимоотношение между линейным программированием и теорией матричных игр и связанные с ним бесконечные итеративные алгоритмы линейного программирования составляют содержание последней главы.

Книга предназначена для математиков, экономистов и инженеров, работающих в области математической экономики, автоматического регулирования и исследования операций. Книга может быть использована также студентами и аспирантами, специализирующимися по вычислительной математике, экономической кибернетике, автоматическому регулированию и исследованию операций.

Монография содержит 13 таблиц, 24 рисунка, 51 библиогр. назв.

*Юдин Давид Беркович
Гольштейн Евгений Григорьевич*

**ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
(теория, методы и приложения)**

М., 1969 г., 424 стр. с илл.

Редакторы: М. М. Горячая и Ю. Л. Заславский

Техн. редактор С. Я. Шкляр

Корректор Г. С. Иванова

Сдано в набор 3/VII 1968 г. Подписано к печати 27/XI 1968 г. Бумага 84×108/32. Физ. печ. л. 13,25. Условн. печ. л. 22,26. Уч.-изд. л. 19,72. Тираж 19 000 экз. Т-15890. Цена книги 1 р. 44 к. Заказ № 2883.

Издательство «Наука».

Главная редакция физико-математической литературы

Москва, В-71, Ленинский проспект, 15.

Ордена Трудового Красного Знамени Первая Образцовая типография имени А. А. Жданова

Главполиграфпрома Комитета по печати при Совете Министров СССР
Москва, Ж-54, Валовая, 28

Предисловие	
-----------------------	--

ГЛАВА I

ВВЕДЕНИЕ

§ 1. Предмет линейного программирования	7
§ 2. Задачи линейного программирования	13
§ 3. Различные формы записи задач линейного программирования	23
§ 4. Геометрия простейших задач линейного программирования	28
§ 5. Экономическая интерпретация задачи линейного программирования	37

ГЛАВА 2

ПРАКТИЧЕСКИЕ ЗАДАЧИ ЛИНЕЙНОГО
ПРОГРАММИРОВАНИЯ

§ 1. О постановке задач линейного программирования	43
§ 2. Применение линейного программирования к определению рациональных норм потребления продуктов питания	46
§ 3. Применение линейного программирования к энергетике	49
§ 4. Применение линейного программирования к металлургии	52
§ 5. Применение линейного программирования к нефтяной промышленности	57
§ 6. Применение линейного программирования в сельском хозяйстве	63
§ 7. Применение линейного программирования к планированию производства	72
§ 8. Применение линейного программирования к планированию экономики	76
§ 9. Об оптимальном раскрое материалов	83

ГЛАВА 3

ЭЛЕМЕНТЫ ТЕОРИИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

§ 1. Необходимые сведения из линейной алгебры и теории выпуклых множеств	85
§ 2. Выпуклые многогранные множества	106
§ 3. Основные свойства задачи линейного программирования	125
§ 4. Геометрия задачи линейного программирования	141
§ 5. Прямая и двойственная задачи	148
§ 6. Теоремы двойственности	163
§ 7. Критерии оптимальности и разрешающие множители	175

ГЛАВА 4

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ ПЛАНА

§ 1. Признак оптимальности	195
§ 2. Общая схема метода	201
§ 3. Геометрические интерпретации метода	210

§ 4. Вырожденность	217
§ 5. Связь между параметрами последовательных итераций	224
§ 6. Первый алгоритм	233
§ 7. Второй алгоритм	244
§ 8. Сравнительная оценка алгоритмов	260
§ 9. Выбор начального опорного плана	263

ГЛАВА 5

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УТОЧНЕНИЯ ОЦЕНОК

§ 1. Основы метода	272
§ 2. Первый алгоритм	286
§ 3. Второй алгоритм	292
§ 4. Способы определения исходного опорного плана сопряженной задачи	297
§ 5. Метод улучшения плана и метод уточнения оценок	302

ГЛАВА 6

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО СОКРАЩЕНИЯ НЕВЯЗОК

§ 1. Общая схема метода	308
§ 2. Алгоритм метода	320
§ 3. Метод двусторонних оценок	327

ГЛАВА 7

НЕКАНОНИЧЕСКАЯ ФОРМА ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

§ 1. Задача линейного программирования в произвольной форме записи	334
§ 2. Естественная и каноническая формы задачи	345
§ 3. Случай двусторонних ограничений	349

ГЛАВА 8

КОНЕЧНЫЕ МЕТОДЫ

§ 1. Классификация конечных методов	358
§ 2. Модификация конечных методов	375

ГЛАВА 9

ИТЕРАТИВНЫЕ МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

§ 1. Теория игр и линейное программирование	390
§ 2. Методы решения игр	396
§ 3. Методы «штрафных функций»	408
Литература	418
Предметный указатель	422

Планирование производства, управление системами и проектирование техники на основе экстремальных принципов экономит время, ресурсы и труд и повышает качество решения экономических и технических задач.

Теоретические основы и методы решения задач планирования, управления и проектирования разрабатываются в новой математической дисциплине, получившей название *математическое программирование*.

Настоящая монография посвящена теории, методам и приложениям наиболее разработанного раздела математического программирования — линейного программирования.

Книга представляет собой переработанное издание монографии тех же авторов «Линейное программирование. Теория и конечные методы», Физматгиз, 1963.

При переработке книги особое внимание уделено приложениям линейного программирования, главным образом анализу экономических задач и модификациям методов и алгоритмов, учитывающим накопленный опыт вычислений.

Книга дополнена изложением итеративных методов решения задач линейного программирования.

В монографию включена глава о различных экономических приложениях общей модели линейного программирования. Интерпретация теории и методов, излагаемых в книге в терминах любой из этих задач, поможет читателю сопоставлять интуитивные эвристические и научные подходы к рациональному планированию.

Книга содержит 9 глав. Глава 1 носит вводный характер. В ней излагаются основные понятия линейного программирования, указывается место этой дисциплины среди других разделов математического программирования и рассматриваются различные методологические вопросы, важные для постановки задач и усвоения методов.

В главе 2 описаны практические приложения общей модели линейного программирования к различным хозяйственным и техническим задачам.

Глава 3 содержит изложение теоретических аспектов линейного программирования. В ней устанавливается связь линейного программирования и теории выпуклых множеств, способствующая усвоению геометрической сущности задачи и методов ее решения. Значительная часть главы 3 посвящена наиболее важному теоретическому вопросу линейного программирования — теории двойственности. Теория двойственности позволяет с единой точки зрения рассматривать различные подходы к построению методов линейного программирования.

Методы линейного программирования, как и методы линейной алгебры, делятся на конечные и итеративные. Главы 4—8 монографии посвящены подробному изложению теории и вычислительных алгоритмов основных конечных методов решения общей задачи линейного программирования. Классификация конечных методов по разным признакам, приведенная в главе 8, позволяет заключить, что изученные в монографии методы можно считать типичными представителями всех существенно различающихся между собой групп конечных методов линейного программирования. В заключительной главе монографии — в главе 9 рассматриваются итеративные методы линейного программирования. В этой же главе исследуется плодотворная связь линейного программирования и теории игр, позволяющая использовать методы одной из этих дисциплин для решения задач другой дисциплины.

Авторы весьма признательны В. В. Боковой за помощь в оформлении рукописи.

Сентябрь 1968

Авторы

§ 1. ПРЕДМЕТ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1. Одной из характерных особенностей современной науки является широкое проникновение результатов и методов математики в самые разнообразные области исследования. Сложные задачи управления народным хозяйством и автоматизации производственных процессов, проблемы планирования экономики и военных операций требуют наряду с изучением качественных закономерностей явлений тщательного исследования их количественных характеристик.

Математические машины, внедряемые в производство и управление и используемые в научно-исследовательской работе, создают огромные возможности для развития различных отраслей науки, для совершенствования методов планирования и автоматизации производства. Однако без строгих формулировок задач, без формально-математического описания процессов не может быть достигнут необходимый уровень использования техники. Зачастую «узким местом» оказываются не столько вопросы, связанные с проектированием машин, сколько вопросы, возникающие в связи с формализацией физических, экономических, технических и других процессов. Формализация задачи — необходимый этап для перевода каждой прикладной задачи на язык математических машин.

Общая схема формализации задач управления и планирования может быть описана следующим образом. Задачи управления и планирования обычно сводятся к выбору некоторых систем параметров или некоторых систем функций. Чтобы иметь основание отдавать предпочтение тем или иным значениям параметров планирования и тем или иным управляющим функциям, необходимо прежде всего четко уяснить себе следующие два

обстоятельства. Во-первых, нужно разобраться в том, как сравнивать между собой различные системы параметров или функций, что следует называть хорошей и наилучшей системой управления или планирования. Иными словами, следует сформулировать и выразить через искомые характеристики показатель качества — критерий, определяющий соответствие разрабатываемых устройств и планов цели, ради которой эта разработка ведется. Во-вторых, необходимо выяснить условия работы системы и вытекающие отсюда ограничения, которым должны удовлетворять искомые параметры и характеристики.

1.2. В весьма широком классе народнохозяйственных, технических, военных и других задач показатель качества выражается линейно через параметры управления или характеристики планирования, а условия, которым должны удовлетворять искомые параметры, записываются в виде линейных равенств или (и) неравенств. Вычисление экстремума (максимума или минимума) линейного показателя качества при условии, что переменные, подлежащие определению, удовлетворяют линейным ограничениям, составляет предмет *линейного программирования*. Актуальность подобных задач в экономике, технике и военном деле, с одной стороны, и вычислительные трудности, связанные с исследованием этих задач, с другой стороны, вызвали к жизни большое количество работ, посвященных различным методам определения экстремума линейных функций при линейных ограничениях на их аргументы и применению этих методов к решению практических задач.

Термин «линейное программирование» нельзя признать удачным, поскольку под словом «программирование» обычно понимают теорию составления программ для универсальных математических машин. Возможно, более удачным названием совокупности вопросов, оформившейся в настоящее время в самостоятельный раздел прикладной математики, было бы «линейное планирование». Однако термин «линейное программирование» уже установился и является общепринятым в литературе. Заметим лишь, что слово «программирование» употребляется здесь в смысле методов (алгоритмов) для решения экст-

1.3. Общая задача линейного программирования формулируется следующим образом. Требуется найти максимум (или минимум) линейной функции n переменных x_1, x_2, \dots, x_n

при ограничениях, наложенных на переменные x_1, x_2, \dots
 \dots, x_n , вида

[illegible]

$$x_1 \geq 0, \dots, x_{n_1} \geq 0 \quad (n_1 \leq n). \quad (1.4)$$

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix},$$

составленную из коэффициентов системы условий (1.2) — (1.4), будем называть *матрицей условий* задачи (1.1) — (1.4).

Система условий задачи (1.1) — (1.4) содержит как равенства, так и неравенства, причем требования неотрицательности, которые обычно выделяют в отдельную систему ограничений, относятся, вообще говоря, к части переменных x_j . Поэтому (1.1) — (1.4) называют иногда задачами со смешанными условиями.

Условия задач линейного программирования иногда бывает удобно записывать в векторном виде. Вектор

$$A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} = (a_{1j}, a_{2j}, \dots, a_{mj})^T,$$

компоненты которого — коэффициенты a_{ij} , стоящие при j -й переменной x_j системы (1.2) — (1.3), назовем j -м вектором условий задачи (1.1) — (1.4).

Вектор

$$B = (b_1, b_2, \dots, b_m)^T,$$

составленный из свободных членов условий (1.2), (1.3), назовем вектором ограничений рассматриваемой задачи. В принятых обозначениях условия (1.2), (1.3) примут вид

$$A_1 x_1 + A_2 x_2 + \dots + A_n x_n \leq B,$$

где знак \leq означает, что равенство относится к первым m_1 компонентам, а неравенство — к остальным $m - m_1$ составляющим.

Набор чисел $X = (x_1, x_2, \dots, x_n)$, удовлетворяющий системе условий (1.2) — (1.4), называется *планом* рассматриваемой задачи. Числа x_1, x_2, \dots, x_n , составляющие план X , будем называть *компонентами* или *составляющими* этого плана.

Каждый план $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ задачи связан с определенным значением $L(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ ее линейной формы. Чем больше это значение (речь идет о задачах, связанных с максимизацией линейной формы), тем «лучше» данный план.

План $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, связанный с максимально возможным значением линейной формы задачи, называется *оптимальным планом* или *решением* задачи. Другими словами, план $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ является оптимальным планом или решением задачи линейного программирования, если

$$L(X^*) \geq L(X) \quad (1.5)$$

для любого плана X задачи.

Естественно, что для задач, состоящих не в максимизации, а в минимизации линейной формы (1.1), условие (1.5) переходит в

$$L(X^*) \leq L(X), \quad (1.6)$$

где неравенство (1.6) должно выполняться для любого плана X задач (1.1)—(1.4).

Термины «план» и «оптимальный план» возникли из экономических приложений линейного программирования (см. § 5). Задача линейного программирования, обладающая хотя бы одним оптимальным планом, называется *разрешимой*.

При анализе системы условий (1.2)—(1.4) могут представиться три случая:

а) условия (1.2)—(1.4) противоречивы; другими словами, не существует набора чисел x_1, x_2, \dots, x_n , удовлетворяющего всем условиям задачи, т. е. задача не имеет ни одного плана;

б) условия (1.2)—(1.4) непротиворечивы, а область, определяемая ими, не ограничена; другими словами, существуют планы со сколь угодно большими по абсолютной величине значениями отдельных компонент;

в) система условий (1.2)—(1.4) совместна, и область, определяемая ею, ограничена.

Неразрешимость задачи может быть обусловлена либо противоречивыми условиями задачи (случай а)), либо неограниченностью области определения линейной формы (случай б)). В последнем случае неразрешимость задачи связана с неограниченностью линейной формы в области ее определения. Заметим, что случай б) не обязательно приводит к неразрешимой задаче. Линейная форма может быть ограничена и в неограниченной области.

1.4. В чем же состоит сложность поставленной задачи? Почему ее анализ требует создания специальных вычислительных методов?

Стандартная схема дифференциального исчисления, используемая обычно для исследования функций на максимум и минимум, применительно к задаче линейного программирования приводит к выделению некоторой конечной системы планов (минимальной системы возможных вариантов), среди которых заведомо содержится искомый оптимальный план (вариант). Далее, согласно классической схеме, следует подвергнуть полученную систему планов дополнительному анализу, который сводится к сравнению значений линейной формы задачи на различных планах системы и завершается отысканием искомого решения. Однако уже при относительно небольших n и m (случаи, к которым сводятся сравнительно простые практические задачи) количество вариантов, составляющих минимальную систему возможных вариантов, исчисляется многими миллиардами. В качестве примера сошлемся на одну из классических моделей линейного программирования — так называемую *задачу выбора*. Суть проблемы выбора в следующем. Задана квадратная таблица с n строками и n столбцами. Требуется выбрать по одному элементу в каждой строке и каждом столбце так, чтобы сумма их оказалась максимальной. Эта задача, имеющая самые разнообразные приложения, оказывается задачей линейного программирования. Минимальная система возможных вариантов задачи насчитывает $n!$ вариантов. Непосредственное решение проблемы выбора связано со сравнением $n!$ величин. Для вычисления каждой из этих величин необходимо произвести n сложений. При $n > 15 - 20$ такое количество операций нельзя провести за обозримый срок на самых быстродействующих современных машинах.

По формуле Стирлинга $n! \approx (n/e)^n \sqrt{2\pi n}$. При $n = 20$ имеем $n! > 2 \cdot 10^{18}$. Машине, выполняющей 1 млрд. операций в секунду (таких машин пока еще нет), потребуется более 500 лет для того, чтобы получить решение этой относительно простой задачи.

При $n = 30$ количество возможных вариантов превышает 10^{31} . По-видимому, легче пересчитать все песчинки

на земном шаре, чем перебрать все эти варианты. Таким образом, отмеченный выше прямой путь решения задачи линейного программирования (построение минимальной системы возможных вариантов и последующий их перебор) — практически невыполнимая работа даже для самых быстродействующих современных вычислительных машин, если даже исследуемая задача имеет сравнительно небольшой размер. Необходимо было построить такие схемы поиска, которые позволили бы выбирать оптимальный план, не перебирая (а следовательно, и не вычисляя) всех возможных вариантов. Эти схемы и составляют содержание линейного программирования.

Как мы увидим в дальнейшем, каждому методу линейного программирования отвечает свой особый способ перебора возможных вариантов, основанный на тех или иных рекомендациях по упорядочиванию процесса поиска искомого решения.

§ 2. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В гл. 2 будут подробно разобраны практические задачи, укладываемые в модель (1.1) — (1.4). Здесь мы намерены лишь кратко ознакомить читателя с приложениями линейного программирования.

В п. 2.1 в задаче об организации снабжения выделены характерные случаи, в которых решение задачи линейного программирования не требует специальных методов и громоздких вычислений. Перевод критерия качества и ограничений на формальный язык в п. 2.1 сразу приводит к задаче вида (1.1) — (1.4). В отличие от этого, задача о размещении заказов и загрузке оборудования, описанная в п. 2.2, сводится к задаче линейного программирования только после некоторых (правда, несложных) искусственных построений. Кроме того, эта задача не может быть решена при помощи элементарных рассуждений, подобных приведенным в п. 2.1. Для анализа задачи о размещении заказов и загрузке оборудования должны быть привлечены специальные методы.

2.1. Организация снабжения. Рассмотрим задачу о рациональной организации снабжения промышленного центра

однородным продуктом, например картофелем или углем. Продукт может привозиться в центр из n пунктов.

Пусть x_i — количество продукта, поставляемое в центр из i -го пункта, а c_i — суммарная стоимость производства и перевозки единицы продукта из i -го пункта. Тогда стоимость продукта, доставленного из i -го пункта в центр, равна $c_i x_i$, а стоимость продукта, привезенного в центр из всех пунктов производства, равна

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n.$$

Требуется организовать снабжение (выбрать x_i) таким образом, чтобы обеспечить минимальную стоимость продукта в центре. При этом необходимо учесть следующие условия: потребность центра в продукте, определяемая величиной b , должна быть удовлетворена; излишков не должно быть. Следовательно, искомые переменные обязаны подчиняться условию

$$x_1 + x_2 + \dots + x_n = b. \quad (2.1)$$

Производство продуктов в i -м пункте ограничено величиной b_i , а пропускная способность транспорта, который может быть использован для перевозки продукта из i -го пункта, ограничена величиной d_i .

Обозначим через β_i меньшее из двух чисел b_i и d_i

$$\beta_i = \min(b_i, d_i), \quad i = 1, 2, \dots, n.$$

Искомые переменные x_i должны, таким образом, удовлетворять ограничениям

$$x_i \leq \beta_i, \quad i = 1, 2, \dots, n. \quad (2.2)$$

Кроме того, значения x_i должны быть неотрицательны, так как перевозки продукта из центра в пункты производства этого продукта исключены:

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \quad (2.3)$$

Если перевозки из i -го пункта могут быть обеспечены двумя видами транспорта, то i -й пункт целесообразно рассматривать как два пункта с разной стоимостью продукта.

Таким образом, мы пришли к задаче линейного программирования. Необходимо обратить в минимум линейную форму L при линейных условиях (2.1) — (2.3).

Сформулированная задача обладает особенностью, позволяющей без труда вычислить значения x_i , при которых будет обеспечена минимальная стоимость продукта в центре. Очевидно, выгодно получать возможно больше продукта из тех пунктов, для которых величина c_i (стоимость производства и перевозки единицы продукта) мала. Перенумеруем пункты производства в порядке возрастания коэффициентов c_i , т. е. предположим, что

$$c_1 \leq c_2 \leq \dots \leq c_n.$$

По условию из первого пункта (откуда поставляется наиболее дешевый продукт) центр может получить не более β_1 единиц продукта. Если общая потребность центра, измеряемая величиной b , не превышает возможностей первого пункта, т. е. $b \leq \beta_1$, то рациональней всего удовлетворить весь спрос только за счет первого пункта. В этом случае

$$x_1 = b, \quad x_2 = x_3 = \dots = x_n = 0.$$

Если же $b > \beta_1$, то целесообразно обеспечить максимально возможную поставку из первого пункта ($x_1 = \beta_1$), а оставшуюся часть потребностей ($b - \beta_1$) удовлетворить за счет других пунктов производства. Мы пришли, таким образом, к задаче организации снабжения, аналогичной предыдущей, с тем лишь различием, что потребность центра определяется теперь величиной $b - \beta_1$, а число пунктов производства равно $n - 1$.

Здесь снова могут представиться два случая. В первом случае $b - \beta_1 \leq \beta_2$. Тогда рациональная система снабжения определяется величинами

$$x_2 = b - \beta_1, \quad x_3 = x_4 = \dots = x_n = 0$$

($x_1 = \beta_1$ определено ранее). Во втором случае, когда $b - \beta_1 > \beta_2$, целесообразно положить $x_2 = \beta_2$ и перейти к новой задаче снабжения с меньшим числом пунктов производства ($n - 2$) и меньшими потребностями центра ($b - \beta_1 - \beta_2$).

Приведенные рассуждения подсказывают следующий путь решения задачи. Из величины b последовательно

вычитаются числа β_1, β_2, \dots . Могут встретиться два случая:

$$a) b - \beta_1 - \beta_2 - \dots - \beta_n > 0;$$

$$б) b - \beta_1 - \beta_2 - \dots - \beta_n \leq 0.$$

В первом случае невозможно полностью удовлетворить потребности центра. Спрос превышает суммарную возможность поставок продукта из всех пунктов производства.

Во втором случае спрос может быть полностью удовлетворен. Определим индекс k из условий

$$b - \beta_1 - \dots - \beta_k \geq 0,$$

$$b - \beta_1 - \dots - \beta_k - \beta_{k+1} < 0.$$

Тогда рациональная система снабжения определяется следующими соотношениями:

$$x_1 = \beta_1, x_2 = \beta_2, \dots, x_k = \beta_k,$$

$$x_{k+1} = b - \beta_1 - \beta_2 - \dots - \beta_k,$$

$$x_{k+2} = \dots = x_n = 0.$$

Таким образом, в оптимальном плане снабжения возможности первых k пунктов используются полностью, а последние $n - (k + 1)$ пунктов в план поставок не включаются вовсе.

Как видим, для решения рассмотренной задачи потребовались лишь элементарные соображения. К сожалению, это не правило, а скорее исключение в задачах линейного программирования. Простота решения здесь является следствием того, что условия задачи содержат только одно ограничение, связывающее все переменные. Все остальные условия ограничивают только область изменения каждой из переменных.

Усложним теперь задачу о снабжении промышленного центра однородным продуктом, включив в нее дополнительное ограничение.

Пусть, например, время, в течение которого транспорт, предназначенный для перевозки продукта, может находиться под погрузкой, ограничено, а механизация погрузочных работ в разных пунктах производства различна.

Обозначим через a_i время, затрачиваемое в i -м пункте на погрузку одной транспортной единицы, а через T — ограничение по сумме времен, затрачиваемых всем транспортом на погрузочные работы. Естественно считать, что количество транспортных единиц y_i , необходимое для перевозки продукта из i -го пункта, пропорционально объему перевозимого продукта. Простой транспорта в i -м пункте производства измеряется величиной $a_i y_i = \alpha a_i x_i$, а суммарный простой транспорта равен

$$\alpha(a_1 x_1 + a_2 x_2 + \dots + a_n x_n).$$

Задача об организации снабжения формулируется теперь следующим образом. Необходимо обратить в минимум линейную форму L при условиях (2.1) — (2.3) и дополнительном ограничении

$$\alpha(a_1 x_1 + a_2 x_2 + \dots + a_n x_n) \leq T.$$

Рассуждения, подобные предыдущим, уже не приводят к решению. В этом случае и тем более при необходимости учета новых ограничений планирование снабжения требует специальных методов, совокупность которых составляет содержание линейного программирования.

Можно, однако, указать случай, когда независимо от количества условий решение задачи линейного программирования достигается элементарными средствами, не требующими трудоемких вычислений. Если область изменения переменных, определяемая условиями задачи, может быть изображена на плоскости (или в трехмерном пространстве), то простые геометрические соображения быстро приводят к решению задачи.

Пусть, например, число пунктов производства в задаче о снабжении равно трем. Придадим для определенности параметрам задачи числовые значения:

$$L = 3x_1 + 5x_2 + x_3, \quad x_1 + x_2 + x_3 = 10, \quad (2.4)$$

$$7x_1 + 4x_2 + 8x_3 \leq 76, \quad (2.5)$$

$$0 \leq x_1 \leq 6, \quad (2.6)$$

$$0 \leq x_2 \leq 3, \quad (2.7)$$

$$0 \leq x_3 \leq 8 \quad (2.8)$$

Введем новые переменные

$$y_1 = x_1, \quad y_2 = L = 3x_1 + 5x_2 + x_3. \quad (2.9)$$

Подставим в (2.4) и (2.9) y_1 вместо x_1 и, пользуясь этими соотношениями, выразим x_2 и x_3 через y_1 и y_2 . После этого, заменив в неравенствах (2.5) — (2.8) x_2 и x_3 полученными для них выражениями, перепишем задачу в новых переменных.

Требуется обратить в минимум линейную форму $L = y_2$ при условиях

$$\begin{aligned} y_2 - y_1 &\geq 14, & 0 \leq y_1 &\leq 6, \\ 0 &\leq y_2 - 2y_1 - 10 &\leq 12, \\ 0 &\leq -y_2 - 2y_1 + 50 &\leq 32. \end{aligned}$$

На плоскости (y_1, y_2) область изменения переменных, ограниченная условиями задачи, представляет собой многоугольник. На рис. 1.1 эта область заштрихована. (Как видно из рисунка, условие $4x_3 = -y_2 - 2y_1 + 50 \geq 0$ оказалось излишним.)

Самая нижняя точка области — вершина A — соответствует наименьшему значению y_2 и, следовательно, представляет собой минимум линейной формы L при заданных ограничениях. Координаты точки A

$$\begin{aligned} y_1 &= 1\frac{1}{3}, \\ y_2 = L &= 15\frac{1}{3}. \end{aligned}$$

Рис. 1.1.

Возвращаясь к исходным переменным, получаем решение задачи

$$x_1 = 1\frac{1}{3}, \quad x_2 = \frac{2}{3}, \quad x_3 = 8, \quad L = 15\frac{1}{3}.$$

Ясно, что такие же геометрические построения без труда приводят к решению аналогичной задачи и при большем количестве ограничений.

Практические задачи, укладываемые в модель линейного программирования, как правило, содержат большое количество переменных и условий. Приведенные интуитивные и наглядные геометрические соображения в общем случае неэффективны. Задачи организации снабжения одного центра неоднородными продуктами или задачи снабжения нескольких центров могут быть быстро решены только при помощи специальных методов.

Рассмотрим, например, задачу об организации снабжения неоднородными продуктами, скажем овощами или топливом. В этом случае могут быть указаны минимальные и предельные потребности в каждом отдельном продукте, например отдельно по картофелю и капусте (или углю и нефти) и условия взаимозаменяемости. Взаимозаменяемость должна быть охарактеризована коэффициентом, указывающим, сколько единиц одного продукта эквивалентно единице другого. Понятие эквивалентности в разных конкретных задачах может определяться по-разному. В рассматриваемой задаче при оценке взаимозаменяемости продуктов можно исходить, например, из калорийности овощей (или топлива).

Сохраняя обозначения, принятые для случая однородного продукта, приведем математическую формулировку задачи.

Требуется обратить в минимум линейную форму

$$L = c_1^{(1)}x_1^{(1)} + \dots + c_n^{(1)}x_n^{(1)} + c_1^{(2)}x_1^{(2)} + \dots + c_n^{(2)}x_n^{(2)}$$

при условиях

$$b_{\min}^{(1)} \leq x_1^{(1)} + \dots + x_n^{(1)} \leq b_{\max}^{(1)},$$

$$b_{\min}^{(2)} \leq x_1^{(2)} + \dots + x_n^{(2)} \leq b_{\max}^{(2)},$$

$$0 \leq x_i^{(1)} \leq \beta_i^{(1)}, \quad 0 \leq x_i^{(2)} \leq \beta_i^{(2)},$$

$$\mu_1 (x_1^{(1)} + \dots + x_n^{(1)}) + \mu_2 (x_1^{(2)} + \dots + x_n^{(2)}) = \mu.$$

Верхний индекс здесь означает номер продукта. Если некоторые из n пунктов являются пунктами производства только одного из рассматриваемых продуктов, то соответствующие значения $x_i^{(s)}$ заранее полагаются равными

нулю. Последнее равенство в системе условий представляет собой условие взаимозаменяемости. Оно может быть истолковано, например, следующим образом: $\mu_1(x_1^{(1)} + \dots + x_n^{(1)})$ — количество калорий, которое можно получить, сжигая $(x_1^{(1)} + \dots + x_n^{(1)})$ тонн угля, $\mu_2(x_1^{(2)} + \dots + x_n^{(2)})$ — теплотворность доставленной нефти, а μ — требуемое количество теплоты. Конечно, не представляет никакого труда обобщение постановки задачи на случай снабжения нескольких центров произвольным количеством неоднородных продуктов.

2.2. Размещение заказов и загрузка оборудования. Рассмотрим задачу о планировании производства сложного оборудования, состоящего из n элементов. Это может быть, например, система, состоящая из n приборов, или прибор из n деталей и т. д.

Заказы могут быть размещены на m различных предприятиях с различным станочным парком и разными возможностями. Требуется выяснить, какое распределение заказов между предприятиями обеспечит производство наибольшего количества комплектных систем в установленный срок. К аналогичным задачам приходим и тогда, когда речь идет о производстве одного прибора, изготовление деталей которого может быть распределено между различными станками предприятия или цеха. Рациональная загрузка оборудования должна обеспечить максимальный выпуск приборов.

Примем за единицу времени время, на которое рассчитана вся программа работы. Пусть в единицу времени на i -м предприятии (на i -м станке) можно изготовить a_{ij} элементов оборудования j -го сорта. Обозначим через x_{ij} время, в течение которого i -й исполнитель (предприятие или станок) загружен изготовлением элементов j -го сорта. Число j -х деталей, выпускаемых всеми исполнителями, равно

$$a_{1j}x_{1j} + a_{2j}x_{2j} + \dots + a_{mj}x_{mj}, \quad j = 1, 2, \dots, n. \quad (2.10)$$

Сдаче подлежат только готовые комплекты продукции. Количество комплектов оборудования, которое можно собрать из изготовленных деталей, лимитируется элементами того сорта, которых произведено меньше всего.

Иными словами, число собранных комплектов L равно наименьшему из чисел (2.10), вычисленных для различных j :

$$L = \min_{1 \leq j \leq n} [a_{1j}x_{1j} + a_{2j}x_{2j} + \dots + a_{mj}x_{mj}]. \quad (2.11)$$

Заказы будут размещены рационально и оборудование будет наилучшим образом загружено, если промежутки времени x_{ij} будут подобраны так, что ни одно предприятие (ни один станок) не будет простаивать, а величина L достигнет максимума. На формальном языке это значит, что необходимо обратить в максимум выражение (2.11) при условиях

$$x_{i1} + x_{i2} + \dots + x_{in} = 1, \quad i = 1, 2, \dots, m, \quad (2.12)$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n. \quad (2.13)$$

Первая группа условий требует, чтобы сумма времен, потраченных каждым исполнителем на изготовление элементов оборудования, равнялась единице, т. е. времени, на которое рассчитана вся программа работы. Это означает, что ни предприятия, ни станки не должны иметь холостого времени.

Вторая группа условий означает, что время загрузки любого исполнителя любым изделием не может быть отрицательным.

Задача на максимум минимума, к которой мы пришли, легко сводится к обычной задаче линейного программирования. Действительно, пусть через V обозначено число комплектов оборудования, которое сможет быть изготовлено из имеющихся деталей.

Поскольку каждый комплект состоит из n различных деталей, то должны иметь место неравенства

$$\left. \begin{aligned} \sum_{i=1}^m a_{i1}x_{i1} &\geq V, \\ \sum_{i=1}^m a_{i2}x_{i2} &\geq V, \\ \dots &\dots \dots \dots \\ \sum_{i=1}^m a_{in}x_{in} &\geq V. \end{aligned} \right\} \quad (2.14)$$

Другими словами, количество $\sum_{i=1}^m a_{ij}x_{ij}$ деталей любого сорта j не должно быть меньше числа комплектов V .

Таким образом, задача оптимальной загрузки оборудования сводится к определению чисел x_{ij} для $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ и числа V , которые удовлетворяют условиям (2.12) — (2.14) и притом доставляют максимальное значение линейной форме

$$L = V, \quad (2.15)$$

зависящей только от одного переменного V .

Мы пришли к задаче линейного программирования. Условия задачи (2.12) — (2.15) могут быть несколько упрощены, если допустить, что каждый станок (предприятие) в состоянии изготавливать детали (элементы оборудования) всех n сортов, т. е.

$$a_{ij} > 0$$

для $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$.

Действительно, нетрудно заметить, что при отмеченных допущениях максимальное количество комплектов оборудования может быть произведено в том случае, когда загрузка исполнителей обеспечивает производство одинакового количества деталей (элементов оборудования) всех сортов, т. е. решение задачи обязательно удовлетворяет условиям

$$\sum_{i=1}^m a_{i1}x_{i1} = \sum_{i=1}^m a_{i2}x_{i2} = \dots = \sum_{i=1}^m a_{in}x_{in}.$$

В самом деле, если бы это было не так и хотя бы одна из этих сумм оказалась больше других, то за счет ее небольшого уменьшения можно было бы увеличить минимальную сумму и повысить таким образом количество комплектов. Но это означало бы, что выбрана не лучшая система размещения заказов и загрузки оборудования.

Следовательно, изложенная задача может быть сформулирована еще и так.

цательность всех переменных. Такое разнообразие форм записи условий задач требует разработки специальных методов для решения отдельных классов задач и затрудняет исследование общих особенностей линейного программирования и создание общих методов и вычислительных алгоритмов. Поэтому естественно рассмотреть способ сведения любой задачи линейного программирования к наиболее простой и удобной для исследования форме.

Будем говорить, что задача линейного программирования записана в *канонической форме*, если она формулируется следующим образом. Найти максимум (минимум) линейной формы

$$L = \tilde{c}_1 x_1 + \dots + \tilde{c}_n x_n \quad (3.1)$$

при условиях

$$\left. \begin{aligned} \bar{a}_{11}x_1 + \dots + \bar{a}_{1n}x_n &= \bar{b}_1, \\ \vdots &\vdots \\ \bar{a}_{m1}x_1 + \dots + \bar{a}_{mn}x_n &= \bar{b}_m, \end{aligned} \right\} \quad (3.2)$$

$$x_1 \geq 0, \dots, x_n \geq 0. \quad (3.3)$$

Условия (3.2) могут быть представлены и в векторном виде

$$\bar{A}_1 x_1 + \bar{A}_2 x_2 + \dots + \bar{A}_n x_n = \bar{B},$$

где \bar{A}_j и \bar{B} — j -й вектор условий и вектор ограничений задачи соответственно.

Задача линейного программирования, записанная в общем виде (1.1) — (1.4), может быть сведена к канонической форме. Введем в задачу (1.1) — (1.4) дополнительные неотрицательные переменные $x_{n+1}, x_{n+2}, \dots, x_{n+(m-m_1)}$ [$x_i \geq 0$ для $i = n+1, \dots, n+(m-m_1)$]. Тогда ограничения (1.2), (1.3) эквивалентны следующим:

$$\begin{array}{llll} a_{11} & x_1 + \dots + a_{1n} x_n & & = b_1, \\ \dots & \dots & \dots & \dots \\ a_{m,1} & x_1 + \dots + a_{m,n} x_n & & = b_{m_1}, \\ a_{m_1+1,1} x_1 + \dots + a_{m_1+1,n} x_n & + x_{n+1} & & = b_{m_1+1}, \\ a_{m_1+2,1} x_1 + \dots + a_{m_1+2,n} x_n & + x_{n+2} & & = b_{m_1+2}, \\ \dots & \dots & \dots & \dots \\ a_{m1} & x_1 + \dots + a_{mn} x_n & + x_{n+m-m_1} & = b_m. \end{array}$$

Дополнительные переменные входят в линейную форму с нулевыми коэффициентами. Если $n_1 = n$, то общая задача линейного программирования уже сведена к канонической форме. В этом случае $\tilde{n} = n + m - m_1$; $\tilde{m} = m$. (Напомним, что n_1 — число неотрицательных переменных в задаче (1.1)—(1.4).)

Пусть теперь смешанные условия сведены к системе \tilde{m} линейных равенств, но $n_1 < n$, т. е. требование неотрицательности относится не ко всем искомым переменным. Простейший путь перехода к канонической форме состоит в замене переменных x_j , не связанных условием неотрицательности, разностью неотрицательных переменных x'_j и x''_j

$$x_j = x'_j - x''_j, \quad (3.4)$$

где

$$x'_j \geq 0, \quad x''_j \geq 0.$$

3.2. Каноническая форма задачи линейного программирования оказывается удобной при построении вычислительных методов линейного программирования. Однако при рассмотрении ряда теоретических вопросов целесообразнее использовать некоторые другие формы записи задач линейного программирования. Различные формы записи задач приходится учитывать также и в вычислительных разделах линейного программирования. Дело в том, что хотя общая схема метода расписывается обычно для канонической формы задачи, его вычислительную реализацию полезно увязывать с естественной формой записи задачи. В большинстве случаев такой подход существенно сокращает трудоемкость численного анализа задачи.

Приведем несколько наиболее употребительных форм записи общей задачи линейного программирования. Задача линейного программирования с *однотипными условиями* содержит только неотрицательные переменные, причем каждое из условий задачи — неравенство. Таким образом, эта форма записи задачи является частным случаем задачи типа (1.1)—(1.4) при

$$n_1 = n, \quad m_1 = 0.$$

Общая задача линейного программирования с однотипными условиями состоит в максимизации (или минимизации) линейной формы

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (3.5)$$

при условиях

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad \text{для } i = 1, 2, \dots, m, \quad (3.6)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (3.7)$$

Условия (3.6) иногда бывает полезно представить в векторном виде

$$A_1x_1 + A_2x_2 + \dots + A_nx_n \leq B,$$

где A_j — j -й вектор условий задачи (3.5)—(3.7), а B — ее вектор ограничений.

В практике линейного программирования встречаются задачи с ограничениями типа неравенств и переменными, которые не предполагаются заранее неотрицательными. Будем называть их задачами, записанными в *сопряженной канонической форме* (смысл этого названия будет разъяснен в гл. 3 при рассмотрении элементов теории двойственности).

Общая задача линейного программирования в сопряженной канонической форме состоит в максимизации (или минимизации) линейной формы

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (3.8)$$

при условиях

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, \quad i = 1, 2, \dots, m. \quad (3.9)$$

Задача (3.8)—(3.9) является частным случаем задачи (1.1)—(1.4) при $n_1 = 0$, $m_1 = 0$. Заметим, что путем объединения систем (3.6), (3.7) в единую систему неравенств задача (3.5)—(3.7) приводится к виду (3.8)—(3.9). Таким образом, любая задача с однотипными условиями может также рассматриваться и как задача, записанная в сопряженной канонической форме.

Произвольная задача линейного программирования вида (1.1)—(1.4) может быть приведена к форме (3.5)—(3.7) или (3.8)—(3.9). Делается это следующим образом. Система уравнений (1.2) разрешается относительно неко-

торых неизвестных; полученные выражения подставляются в линейную форму (1.1) задачи и в системы неравенств (1.3), (1.4). В результате задача принимает вид (3.8)—(3.9). Если затем освободиться от тех переменных, которые не связаны условиями неотрицательности, то приходим к задаче с однотипными условиями. Последняя операция осуществляется, например, в соответствии с формулой (3.4).

3.3. В ряде приложений возникает следующая задача, обобщающая задачу (1.1)—(1.4). Рассмотрим l различных линейных форм

$$c_{s1}x_1 + c_{s2}x_2 + \dots + c_{sn}x_n, \quad s = 1, 2, \dots, l.$$

Для каждого набора переменных

$$X = (x_1, x_2, \dots, x_n)$$

определим функцию

$$V(X) = \min_{1 \leq s \leq l} (c_{s1}x_1 + c_{s2}x_2 + \dots + c_{sn}x_n), \quad (3.10)$$

равную наименьшему из l чисел

$$c_{s1}x_1 + c_{s2}x_2 + \dots + c_{sn}x_n, \quad s = 1, 2, \dots, l.$$

Задача состоит в максимизации функции $V(X)$ при условиях (1.2)—(1.4). Пример практической задачи, приводящей к максимизации функции вида (3.10) при линейных ограничениях на ее переменные, был описан в п. 2.2.

Если $l = 1$, то сформулированная задача превращается в задачу (1.1)—(1.4). Покажем, что при любом значении l данная задача приводится к модели линейного программирования. Введем новую переменную V , которую свяжем с переменными x_j условиями

$$a_{s1}x_1 + a_{s2}x_2 + \dots + a_{sn}x_n \geq V, \quad s = 1, 2, \dots, l. \quad (3.11)$$

Эквивалентная задача линейного программирования состоит в максимизации линейной формы

$$V \quad (3.12)$$

при условиях (3.11), (1.2)—(1.4), наложенных на переменные x_1, x_2, \dots, x_n, V . Для того чтобы убедиться в экви-

валентности задач (3.10), (1.2)—(1.4) и (3.12), (3.11), (1.2)—(1.4), достаточно заметить, что:

а) при любых x_1, x_2, \dots, x_n , подчиняющихся условиям (1.2)—(1.4), набор чисел

$$x_1, x_2, \dots, x_n, V(x_1, x_2, \dots, x_n)$$

— план задачи (3.12), (3.11), (1.2)—(1.4);

б) если $x_1^*, x_2^*, \dots, x_n^*, V^*$ — решение задачи (3.12), (3.11), (1.2)—(1.4), то

$$V^* = V(x_1^*, x_2^*, \dots, x_n^*).$$

Таким образом, задача (3.10) с условиями (1.2)—(1.4) при произвольном l является задачей линейного программирования и, следовательно, может быть приведена к любому из видов, рассмотренных в этом параграфе.

Задача (3.10), (1.2)—(1.4) была сформулирована Л. В. Канторовичем (см., например [18]). В дальнейшем мы будем называть ее иногда задачей линейного программирования в форме Л. В. Канторовича.

§ 4. ГЕОМЕТРИЯ ПРОСТЕЙШИХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

4.1. Рассмотрим задачи линейного программирования, допускающие геометрическое истолкование и решение на плоскости и в пространстве.

Начнем с задачи линейного программирования с двумя переменными x_1, x_2 и m однотипными условиями-неравенствами

$$c_1x_1 + c_2x_2 \rightarrow \max(\min), \quad (4.1)$$

$$a_{i1}x_1 + a_{i2}x_2 \leq b_i, \quad i = 1, 2, \dots, m, \quad (4.2)$$

$$x_j \geq 0, \quad j = 1, 2. \quad (4.3)$$

Допустим, что $c_1c_2 \neq 0$, т. е. линейная форма (4.1) не вырождается в постоянное число. Геометрическое место точек (x_1, x_2) , в которых линейная форма (4.1) принимает фиксированное значение L , определяется уравнением

$$c_1x_1 + c_2x_2 = L, \quad (4.4)$$

т. е. является прямой линией, перпендикулярной вектору $n = (c_1, c_2)$. Придавая параметру L всевозможные значения от $-\infty$ до ∞ , получаем семейство параллельных прямых, заполняющее всю плоскость $x_1 O x_2$. Переход от одной прямой семейства к другой в направлении, определяемом вектором $n = (c_1, c_2)$, приводит к возрастанию линейной формы (4.1); противоположное движение ведет к убыванию этой формы.

Итак, линейная форма (4.1) может интерпретироваться геометрически в виде прямой (4.4), вдоль которой она сохраняет постоянное значение, и вектора $n = (c_1, c_2)$, указывающего направление возрастания этой формы.

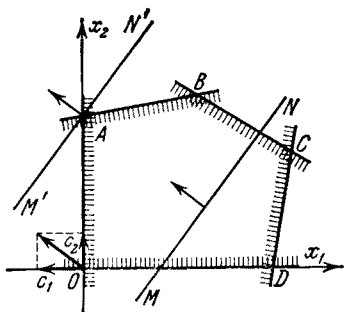


Рис. 1.2.

На рис. 1.2 изображен случай, когда область изменения линейной формы представляет собой многоугольник. Прямые, отсекающие многоугольник $OABCD$ на плоскости $x_1 O x_2$, определяются условиями (4.2) при $m=3$ и (4.3), в которых знаки неравенства заменены на знаки равенства. Штриховка указывает ту сторону прямой, по которую располагаются точки плоскости, удовлетворяющие неравенствам (4.2) и (4.3) (нижние полуплоскости соответствующих прямых). Направление прямой MN определяется вектором $n = (c_1, c_2)$ (вектор n перпендикулярен прямой MN). Задача линейного программирования — вычисление координат точки, дающей экстремум линейной форме (4.1) при условиях (4.2) и (4.3), — может быть геометрически истолкована следующим образом.

Пересечем многоугольник условий — область определения линейной формы — прямой $L = c_1 x_1 + c_2 x_2$ и будем перемещать прямую параллельно самой себе в направлении увеличения L (если требуется вычислить максимум линейной формы) или в направлении уменьшения L (если требуется вычислить минимум линейной формы).

Если исследуемая задача состоит в максимизации (минимизации) линейной формы, то перемещение прямой осуществляется до такого предельного положения, когда многоугольник условий содержится в нижней (верхней) полуплоскости, порождаемой соответствующей предельной прямой, причем хотя бы одна его точка принадлежит этой прямой. Полученная предельная прямая носит название *опорной прямой многоугольника условий*. В случае,

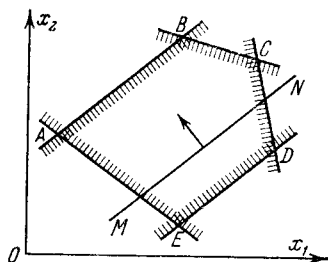


Рис. 1.3.

изображенном на рис. 1.2, параллельный сдвиг в сторону максимизации L приведет прямую MN в такое предельное положение $M'N'$, когда у нее окажется только одна общая точка с многоугольником — вершина A . Эта точка определяет единственное решение задачи линейного программирования (на рис. 1.2 максимум линейной формы). Может оказаться (такой случай изображен на рис.

1.3), что прямая MN параллельна одной или двум сторонам многоугольника. В таком случае экстремум достигается во всех точках соответствующей стороны многоугольника. На рис. 1.3 во всех точках стороны AB многоугольника $ABCDE$, параллельной прямой MN , достигается максимум, а во всех точках стороны $ED \parallel MN$ достигается минимум линейной формы. Таким образом, задача линейного программирования может иметь либо одно, либо бесконечное количество решений. Из рис. 1.3 следует, что если две вершины дают экстремум линейной формы, то и все точки отрезка, соединяющего эти вершины, определяют решение задачи линейного программирования.

Рис. 1.4 соответствует случаю, когда задача линейного программирования неразрешима, поскольку определяющие ее условия оказались противоречивыми.

На рис. 1.5 область определения линейной формы не ограничена. В том случае, когда прямая $AB \parallel MN$, линейная форма достигает конечного экстремума во всех точках луча AB . Если изменять область определения

линейной формы, поворачивая луч AB относительно точки A , то можно получить два случая: линейная форма может либо стать неограниченной при допустимых значениях переменных, либо достигнуть максимума в единственной точке. Первый случай соответствует лучу AB' , изображенному на рис. 1.5 пунктиром. Второму случаю соответствует штрих-пунктирный луч AB'' на том же рисунке.

Такое же наглядное геометрическое истолкование задачи линейного программирования имеет место для

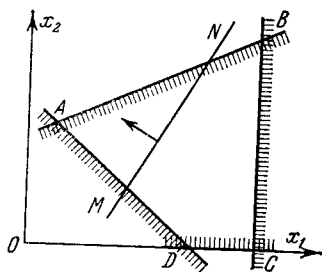


Рис. 1.4.

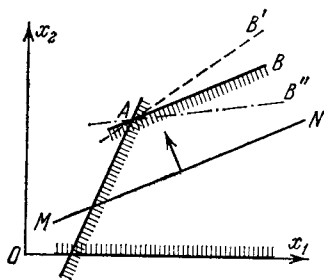


Рис. 1.5.

случая трех переменных. Условия задачи высекают в пространстве выпуклый многогранник (или выпуклое многогранное множество). Коэффициенты линейной формы определяют семейство параллельных плоскостей N и направление, в котором увеличивается L .

Для решения задачи линейного программирования следует перемещать плоскость N , пересекающую многогранник, в сторону увеличения линейной формы (если решается задача на максимум) или в сторону уменьшения L (если решается задача на минимум) до тех пор, пока она еще содержит точки многогранника. Предельное положение плоскости определяет решение задачи.

Геометрические соображения подсказывают здесь, как и прежде, что экстремум достигается в крайних точках — в вершинах многогранника. Если экстремум достигается более чем в одной точке, то он достигается на всем

ребре или на всей грани многогранника, параллельных плоскости, определяемой коэффициентами линейной формы задачи.

4.2. Геометрическое описание задачи линейного программирования, изложенное в предыдущих пунктах, носит название ее *первой геометрической интерпретации*.

Приведем еще одно геометрическое истолкование задач линейного программирования. *Вторая геометрическая интерпретация* приспособлена для задач, записанных в канонической форме. Мы здесь рассмотрим задачи с двумя условиями-равенствами и произвольным числом переменных. Вторая геометрическая интерпретация дает наглядное истолкование таких задач в трехмерном пространстве.

Рассмотрим задачу линейного программирования, состоящую в максимизации линейной формы

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (4.5)$$

при условиях

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \end{aligned} \right\} \quad (4.6)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (4.7)$$

Введем новые переменные:

$$\left. \begin{aligned} u_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n, \\ u_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n, \\ u_3 &= c_1x_1 + c_2x_2 + \dots + c_nx_n. \end{aligned} \right\} \quad (4.8)$$

Соотношения (4.8) позволяют для каждой системы n чисел x_1, x_2, \dots, x_n (n -мерного вектора) однозначно определить тройку чисел u_1, u_2, u_3 .

Введем в трехмерном пространстве прямоугольную систему координат с началом в точке O и осями Ou_1, Ou_2, Ou_3 . В этой системе любому вектору $X = (x_1, x_2, \dots, x_n)$ однозначно сопоставляется точка с координатами u_1, u_2, u_3 , которые вычисляются по формулам (4.8). Каждому множеству векторов отвечает некоторое мно-

жество точек трехмерного пространства. Определим, какому множеству отвечает совокупность всевозможных векторов $X = (x_1, x_2, \dots, x_n)$ с неотрицательными компонентами ($x_j \geq 0$ для всех значений j).

Пусть x_1 изменяется от 0 до ∞ , в то время как $x_2 = x_3 = \dots = x_n = 0$. Из соотношений (4.8) вытекает, что в трехмерном пространстве переменных u_1, u_2, u_3 указанной совокупности векторов X отвечает луч Λ_1 , выходящий из начала координат и идущий вдоль вектора (a_{11}, a_{21}, c_1) . Аналогично при любом j множеству векторов $X = (x_1, x_2, \dots, x_n)$, у которых все компоненты, кроме j -й, равны нулю, а x_j изменяется от 0 до ∞ , соответствует луч Λ_j , исходящий из начала координат и направленный вдоль вектора (a_{1j}, a_{2j}, c_j) .

Из формул (4.8) непосредственно следует, что образ (u_1, u_2, u_3) произвольного вектора $X = (x_1, x_2, \dots, x_n)$ с неотрицательными компонентами может быть представлен в виде суммы некоторых векторов, расположенных на лучах Λ_j ; ими являются векторы

$$(x_j a_{1j}, x_j a_{2j}, x_j c_j), \quad j = 1, 2, \dots, n.$$

С другой стороны, в результате сложения любого числа векторов, расположенных на лучах Λ_j , получаем образ некоторого вектора $X = (x_1, x_2, \dots, x_n)$ с неотрицательными составляющими. Таким образом, искомое множество представляет собой совокупность сумм всевозможных векторов, расположенных на лучах Λ_j . Геометрические соображения подсказывают, что это множество является выпуклым многогранным конусом, т. е. конусом, который вместе с любыми двумя векторами содержит их сумму и граница которого состоит из кусков конечного числа плоскостей.

Обозначим полученный выпуклый многогранный конус через K . Этот конус изображен на рис. 1.6. Вершина конуса K расположена в начале координат; каждое из его ребер совпадает с одним из лучей Λ_j . Отметим, что некоторые лучи Λ_j могут и не являться ребрами конуса K . Поэтому количество его ребер меньше либо равно n . рис. 1.6 соответствует случаю, когда каждый луч Λ_j — ребро конуса K . Конус K порождается лучами Λ_j , которые

в свою очередь определяются векторами

$$\bar{A}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ c_j \end{pmatrix}.$$

Векторы \bar{A}_j принято называть *расширенными векторами условий*. Расширенный вектор условий \bar{A}_j образуется путем добавления к вектору условий A_j еще одной компо-

ненты, равной коэффициенту c_j линейной формы задачи,

В рассматриваемом случае расширенный вектор условий содержит три компоненты. В общем случае расширенный вектор условий \bar{A}_j содержит $m+1$ компонент и вводится аналогичным образом

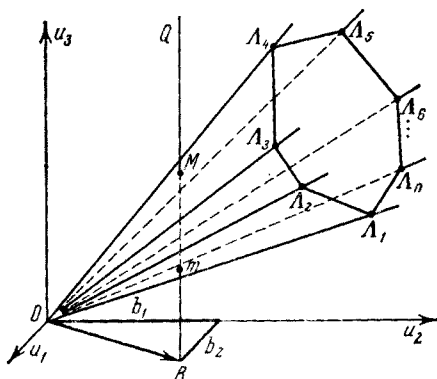


Рис. 1.6.

$$\bar{A}_j = \begin{pmatrix} A_j \\ c_j \end{pmatrix}.$$

Выпуклый многогранный конус K является образом в трехмерном пространстве точек (u_1, u_2, u_3) множества векторов X , удовлетворяющих условиям (4.7).

Проследим теперь, во что переходят векторы X , подчиняющиеся ограничениям (4.6). Для таких векторов

$$u_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1,$$

$$u_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2.$$

Естественно предположить, что не существует чисел λ_1 и λ_2 , при которых

$$c_j = \lambda_1 a_{1j} + \lambda_2 a_{2j}$$

для $j = 1, 2, \dots, n$.

В противном случае для любого вектора $X = (x_1, x_2, \dots, x_n)$, удовлетворяющего условиям (4.6),

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = \lambda_1(a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n) + \\ + \lambda_2(a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n) = \lambda_1b_1 + \lambda_2b_2,$$

т. е. линейная форма задачи (4.5) — (4.7) сохраняет постоянное значение на множестве своих планов. Такие задачи не представляют интереса и не будут здесь рассматриваться.

Из сделанного допущения вытекает, что при любом q найдутся такие x_1, x_2, \dots, x_n , что

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ c_1x_1 + c_2x_2 + \dots + c_nx_n = q.$$

Итак, образом множества векторов X , удовлетворяющих условиям (4.6), является прямая Q , для которой

$$u_1 = b_1, \quad u_2 = b_2, \quad u_3 = q, \quad -\infty < q < \infty.$$

Прямая Q параллельна оси Ou_3 и проходит через точку (b_1, b_2) плоскости u_1Ou_2 (см. рис. 1.6). Каждый план задачи (4.5) — (4.7) должен удовлетворять как условиям (4.6), так и условиям (4.7). Поэтому области определения задачи (4.5) — (4.7) отвечает общая часть выпуклого многогранного конуса K и прямой Q . Если прямая Q не имеет общих точек с конусом K , то рассматриваемая задача не имеет ни одного плана. В этом случае она, естественно, неразрешима. Допустим, что задача (4.5) — (4.7) имеет планы и, следовательно, общая часть конуса K и прямой Q содержит по крайней мере одну точку. В силу выпуклости K общая часть K_Q конуса K и прямой Q представляет собой отрезок, луч либо прямую (отрезок может вырождаться в точку). В случае, изображенном на рис. 1.6, множество K_Q — отрезок с концами m и M .

Пусть (b_1, b_2, q) — произвольная точка, расположенная на K_Q . Эта точка соответствует планам задачи (4.5) — (4.7), на которых значение линейной формы (4.5) равно q . Следовательно, решение задачи сводится к определению одного из концов K_Q (верхнего для задач максимизации

и нижнего для задач минимизации). В рассматриваемом случае необходимо определить верхнюю точку пересечения прямой Q с конусом K . Если этой точки не существует (K_Q — луч, направленный вдоль положительной полуоси Ou_3 , или прямая), то задача (4.5) — (4.7) неразрешима из-за неограниченности линейной формы (4.5) на множестве планов. На рис. 1.6 изображен случай разрешимой задачи. Ее решению отвечает точка M — верхняя точка пересечения конуса K , образованного лучами $\Lambda_1, \Lambda_2, \dots, \Lambda_n$, и прямой Q . Для того чтобы определить составляющие решения, выделим грань, в которой расположена точка M . Из рис. 1.6 видно, что эта грань образована лучами Λ_3 и Λ_4 .

Следовательно,

$$M = \begin{pmatrix} b_1 \\ b_2 \\ q_{\max} \end{pmatrix} = x_3^* \begin{pmatrix} a_{13} \\ a_{23} \\ c_3 \end{pmatrix} + x_4^* \begin{pmatrix} a_{14} \\ a_{24} \\ c_4 \end{pmatrix},$$

и оптимальным планом является вектор

$$X^* = (0, 0, x_3^*, x_4^*, 0, \dots, 0),$$

где x_3^* и x_4^* определяются в результате решения системы уравнений

$$a_{13}x_3^* + a_{14}x_4^* = b_1,$$

$$a_{23}x_3^* + a_{24}x_4^* = b_2.$$

В данном случае задача имеет единственное решение. Отметим, что неединственность решения задачи (4.5) — (4.7) возможна только тогда, когда грань конуса K , в которой расположена точка M , содержит более двух лучей Λ_j .

Таким образом, вторая геометрическая интерпретация задачи линейного программирования позволяет провести наглядный геометрический анализ задач типа (4.5) — (4.7).

Если увеличивать число переменных для задач с одно-типными условиями или количество ограничений-равенств для задач в канонической форме, то наглядность геометрических построений теряется, так как их следует проводить теперь в r -мерных пространствах при $r > 3$. Для анализа задач линейного программирования, не уклады-

вающихся в класс простейших задач, рассмотренных в этом параграфе, разработаны специальные аналитические методы, изложению которых посвящена большая часть книги.

§ 5. ЭКОНОМИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

5.1. Основной областью приложения линейного программирования являются экономические задачи. Собственно линейное программирование и возникло из потребностей экономики. Поэтому целесообразно наряду с геометрическим истолкованием задачи линейного программирования ввести также ее экономическую интерпретацию.

Рассмотрим модель планирования работы предприятия или группы предприятий, производящих некоторый однородный продукт. Производство продукта требует ряда категорий сырья, определенного станочного парка, рабочей силы различной квалификации, энергии, топлива, транспорта и т. д. Будем называть перечисленные факторы *производственными*. Пусть число факторов, необходимых для производства продукта, равно m . Как правило, любой из них имеется на предприятиях в ограниченном количестве.

В каждой конкретной задаче можно указать много технологических способов производства, различным образом использующих производственные факторы для производства продукции. Технологический способ производства однородного продукта можно характеризовать системой $m + 1$ чисел, определяющих затраты каждого производственного фактора при использовании его в течение единицы времени и выпуск продукции за то же время. Можно себе представить следующий порядок составления плана работы предприятия.

Заранее отрабатывается (или предполагается заданным) ряд технологических способов производства — ряд наборов затрат каждого из производственных факторов в единицу времени и обеспечиваемый ими выпуск продукции. В зависимости от того, сколько времени предприятие работает в соответствии с тем или иным из исходных технологических способов производства, будет затрачиваться

различное количество производственных факторов разных категорий и производится различное количество продукции. Задача планирования производства состоит в определении продолжительности работы по отдельным технологическим способам производства (или в определении интенсивности использования каждого из них).

Наилучшим планом естественно признать тот план, при котором затраты по каждому производственному фактору не превысят допустимых, а продукция предприятия достигнет наибольшего возможного значения. Очевидно, чем больше проанализировано и отработано исходных способов производства, тем лучшим будет оптимальный план.

Формализация задачи планирования предприятия в указанном выше смысле требует следующих вполне естественных в широком круге вопросов предположений. Если один набор затрат производственных факторов обеспечивает в единицу времени c_1 единиц продукции, а другой c_2 , то работа предприятия в течение x_1 единиц времени в соответствии с первым набором затрат и x_2 согласно второму набору обеспечит $c_1x_1 + c_2x_2$ единиц продукции. Конечно, нет никакой необходимости предполагать, что промежутки времени x_1 и x_2 следуют один за другим. Предприятие может одновременно использовать различные технологические способы производства, заменяя в определенные моменты времени отдельные способы производства на другие. Промежутки времени x_1 и x_2 , таким образом, играют здесь роль удельных весов (интенсивностей), с которыми исходные способы участвуют в выбранном для предприятия технологическом способе производства. Время перехода с одного технологического способа производства на другой — время холостого хода — в рассматриваемой постановке задачи не учитывается.

5.2. Переведем задачу о планировании работы предприятия на математический язык.

Обозначим через c_j количество единиц продукции, выпущенной предприятием при работе в течение единицы времени по j -му технологическому способу производства, когда затраты 1-го производственного фактора составляют a_{1j} , 2-го — a_{2j} и т. д. ..., j -го — a_{ij} , ..., и, наконец,

m -го — a_{mj} . Таким образом, j -й технологический способ производства определяется набором $m+1$ чисел a_{1j} , a_{2j} , ..., a_{ij} , ..., a_{mj} , c_j . Пусть, кроме того, числа b_1 , b_2 , ..., b_m определяют наличные ресурсы разных производственных факторов. Величины b_i ограничивают размеры допустимых затрат.

Будем предполагать, что, приступая к планированию, мы располагаем n отработанными способами производства. Обозначим через x_1 , x_2 , ..., x_n промежутки времени, в течение которых предприятие работает соответственно по 1-му, 2-му, ..., n -му способу производства. Суммарная продукция, выпущенная предприятием, определяется при этом выражением

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n. \quad (5.1)$$

Ограничения по расходам отдельных производственных факторов записываются в виде следующих условий:

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n \leq b_i, \quad i = 1, 2, \dots, m. \quad (5.2)$$

В левой части каждого из неравенств системы (5.2) стоят суммарные расходы соответствующего производственного фактора при данном плане использования различных технологических способов, в правой — ограничение по этому фактору.

Естественно, что промежутки времени использования различных технологических способов производства неотрицательны, т. е.

$$x_j \geq 0 \quad \text{для} \quad j = 1, 2, \dots, n. \quad (5.3)$$

Как видим, задача планирования производства свелась к задаче линейного программирования с однотипными условиями. Выбор плана в рассмотренной задаче — это выбор системы чисел x_1 , x_2 , ..., x_n , при которой удовлетворяются условия (5.2) и (5.3), а линейная форма (5.1) достигает своего наибольшего значения.

5.3. Итак, задача линейного программирования с однотипными условиями может интерпретироваться как задача составления оптимального плана использования существующих технологических способов производства, который при заданных запасах различных производственных

факторов приводит к максимально возможному объему конечной (однородной) продукции. Естественно, что все параметры c_j , a_{ij} , b_i задачи предполагаются при этом неотрицательными числами.

Экономическая интерпретация задачи линейного программирования впоследствии окажется полезной для интуитивного обоснования и, следовательно, более глубокого уяснения различных понятий линейного программирования. Эта интерпретация позволит также перефразировать в естественных экономических терминах ряд теоретических результатов и вычислительных алгоритмов линейного программирования, что, несомненно, должно способствовать неформальному усвоению излагаемых фактов.

Следует отметить, что приведенная здесь интерпретация задачи линейного программирования является далеко не единственно возможным способом описания задачи в терминах конкретной прикладной постановки. Можно было бы указать целый ряд других экономических (да и не только экономических) моделей, в терминах которых формулировка общей задачи линейного программирования оказывается столь же естественной. Некоторые из этих моделей описаны в гл. 2. Вместе с тем соображения удобства изложения привели к необходимости остановиться на какой-то одной модели, в терминах которой можно было бы пояснять смысл различных понятий, результатов и алгоритмов на протяжении всей книги. В качестве такой модели выбрана задача производственного планирования — одно из наиболее важных приложений линейного программирования.

В связи с приведенной экономической интерпретацией целесообразно остановиться на некоторых элементарных понятиях линейного программирования, введенных в предыдущих параграфах.

Векторы

$$A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$$

и

$$B = (b_1, b_2, \dots, b_m)^T$$

были названы ранее j -м вектором условий и вектором

ограничений соответственно. В экономической интерпретации компоненты некоторого вектора условий определяют затраты различных производственных факторов в единицу времени при использовании соответствующего технологического способа производства, а компоненты вектора ограничений — запасы отдельных факторов, ограничивающие их расход. В терминах производственной задачи векторы A_j и B естественнее называть соответственно *векторами затрат* и *вектором запасов*.

Как уже отмечалось, j -й технологический способ производства определяется вектором \bar{A}_j , первые m компонент которого совпадают с составляющими вектора условий A_j , а последняя составляющая равна c_j . Векторы \bar{A}_j были названы ранее расширенными векторами условий. Экономическая интерпретация подсказывает для них другое название: \bar{A}_j — *вектор j -го технологического способа производства*.

В ряде случаев удобно использовать понятие расширенного вектора ограничений \bar{B} , первые m компонент которого совпадают с составляющими вектора ограничений B , а последняя равна экстремальному значению линейной формы задачи. Таким образом, вектор \bar{B} определяет запасы всех ресурсов и наибольшее значение продукции, которое может быть получено при данных условиях. Поэтому расширенный вектор ограничений естественно называть *вектором возможностей производства*.

В связи с рассмотренной экономической интерпретацией становится понятным происхождение таких терминов, как план и оптимальный план, введенных в предыдущих параграфах. Линейное программирование возникло из экономических приложений, поэтому исходные понятия этой дисциплины часто формулировались в связи с той или иной конкретной экономической задачей. Термины план и оптимальный план были первоначально введены как понятия, относящиеся соответственно к распределению и оптимальному распределению имеющихся ресурсов.

Рассмотренная здесь экономическая интерпретация относится к задаче линейного программирования с однотипными условиями. Столь же естественное экономическое истолкование можно дать и для других форм записи

задачи. Например, задача линейного программирования в форме Л. В. Канторовича соответствует проблеме составления оптимального плана использования имеющихся технологических способов с тем, чтобы выпустить как можно больше продуктов различных наименований в заданном ассортименте, не выходя при этом из ограничений, диктуемых существующими запасами по различным факторам производства. Более подробно эта интерпретация разбирается в § 7 гл. 2.

Учитывая, однако, что условия любой линейной задачи могут быть легко сведены к однотипному виду, можно считать приведенное здесь экономическое истолкование задачи применимым и в общем случае.

ПРАКТИЧЕСКИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В этой главе рассматриваются разнообразные экономические и технические модели, анализ которых сводится к решению задач линейного программирования общего вида. Всюду, где это необходимо, подчеркиваются особенности модели, существенные для формализации задачи.

§ 1. О ПОСТАНОВКЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Математический анализ применяется не к реальным явлениям, а к некоторым математическим моделям этих явлений. Такие абстрактные модели, естественно, охватывают не все, а лишь важнейшие для данной задачи стороны явления. Наиболее квалифицированная и ответственная работа при постановке задачи заключается в выборе характеристик явления, существенных для данной задачи и подлежащих формализации и включению в математическую модель.

Изучаемые явления не изолированы. Они связаны и взаимодействуют с другими явлениями природы и жизни, возможно, не представляющими интереса для рассматриваемой задачи. При постановке задачи следует решить, какими связями можно пренебречь и какие связи следует заменить теми или иными ограничениями на выбор искомых параметров. В зависимости от того, насколько тщательно проведена эта часть работы, определяется целесообразность применения сложного математического аппарата для анализа задачи и практическая польза от ее решения.

При формализации задачи управления и планирования следует уделить особое внимание выбору показателя качества решения соответствующей задачи линейного

программирования. Обычно к искомому методу управления или плану предъявляют самые разнообразные, порой противоречивые требования. Как правило, нельзя указать план (или метод управления), на котором достигались бы экстремумы различных показателей. Анализ целей, для достижения которых проводится планирование, должен выявить важнейший показатель, подлежащий оптимизации, и допустимые границы изменения остальных показателей. Оптимизируемая характеристика — это показатель качества решения. Допустимые границы изменения других характеристик решения определяют ограничения задачи.

Важным этапом в постановке задачи является выбор переменных, которые следует принять в качестве параметров управления. При выборе переменных желательно обеспечить возможно более простой вид условий и показателя качества. От удачного выбора параметров управления существенно зависит трудоемкость решения задачи.

При построении математических моделей исследуемых явлений, особенно в тех случаях, когда эти явления изучаются впервые, не всегда удастся сразу сформулировать и записать все условия, которые должны ограничивать область изменения переменных задачи. Некоторые факторы и ограничения, представляющиеся естественными, предполагаются само собой подразумевающимися и специально не оговариваются. Если решение задачи производится не формально, то рассуждения в терминах конкретного приложения обычно приводят к тому, что задача в процессе решения обрастает дополнительными ограничениями, вытекающими из ее физической сущности. Однако неформальное решение задачи возможно лишь при весьма малом числе переменных и ограничений.

Задачи линейного программирования, к которым сводятся практические проблемы планирования и управления, как правило, содержат большое число переменных и ограничений и далеко не всегда допускают умозрительный анализ. При исследовании таких задач необходима тщательная формализация задачи и включение в математическую модель всех сколько-нибудь существенных ограничений, какими бы тривиальными они ни представлялись.

Необходимо специально оговаривать неотрицательность переменных для параметров, которые по своему

физическому смыслу не могут быть меньше нуля. Известны примеры, в которых абсурдность формально полученных решений объясняется отсутствием среди формально записанных условий задачи указаний о запрещенной области изменения переменных. Известен также случай, когда при решении задачи о наиболее дешевой диете, обеспечивающей необходимые питательные вещества в требуемых количествах, были получены совершенно несъедобные рационы. Такой результат явился следствием того, что при составлении математической модели задачи не были учтены вкусовые характеристики диеты. Иногда встречаются случаи, когда при анализе задач линейного программирования получают практически нереализуемые решения: при формализации задачи не были зафиксированы все лимитирующие факторы, от которых зависит выбор искомого переменных. Часто необходимость учета дополнительных ограничений выясняется только после анализа причин нереализуемости решения задачи.

Процесс постановки и решения практических задач не всегда удается провести в один этап. Обычно количественный анализ решения указывает направление, в котором следует уточнить модель, чтобы она полнее отражала реальное явление. После предварительной постановки задачи целесообразно получить ее формальное решение для простейших случаев, для которых оптимальный план может быть получен (точно или приближенно) из физических соображений. Анализ отклонения формального решения от ожидаемого результата позволяет скорректировать постановку задачи или (это тоже иногда встречается) меняет наши представления о том, что следует считать естественным. При сравнении формальных решений с предполагаемыми результатами следует, однако, иметь в виду, что факторы, существенные в одном диапазоне изменения исходных параметров задачи, могут не играть сколько-нибудь заметной роли в другой области их изменения.

Учет дополнительных факторов, выявленных в процессе постановки задачи, может заставить изменить выбор параметров управления, подлежащих определению. Сколь угодно подробный учет ограничений на искомые переменные не приближает решения проблемы, если

построенная математическая модель не может быть исследована существующими методами. Поэтому постановку серьезных задач линейного программирования целесообразно проводить специалистам прикладных наук совместно с математиками.

§ 2. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К ОПРЕДЕЛЕНИЮ РАЦИОНАЛЬНЫХ НОРМ ПОТРЕБЛЕНИЯ ПРОДУКТОВ ПИТАНИЯ

Задача о диете является одним из первых применений линейного программирования к практическим потребностям. Эта задача в различных формулировках применительно к самым разнообразным приложениям часто обсуждалась в периодической литературе. Задача ставится по-разному в зависимости от того, предполагается воспользоваться решением для составления перспективного плана развития пищевой промышленности или для планирования централизованного снабжения продуктами питания детских учреждений, пионерских лагерей, военных частей, специализированных больниц и т. д.

Приведем простейшую постановку задачи об определении рациональных норм потребления продуктов питания. Учет специфики конкретных приложений позволяет посредством дополнительных ограничений или путем модификации показателя качества решения приспособить эту модель к особенностям соответствующей задачи.

Различные пищевые продукты содержат питательные вещества и витамины необходимых типов в определенных, но различных пропорциях. Минимальные потребности в тех или иных питательных веществах и витаминах известны. Зная величину запасов различных продуктов и стоимость каждого из них, необходимо определить, как можно удовлетворить потребности населения при минимальных затратах.

Приведем математическую формулировку этой задачи. Пусть имеется n различных продуктов P_1, P_2, \dots, P_n . Обозначим через x_j суточное потребление j -го продукта. Диета характеризуется, таким образом, системой чисел $X = (x_1, x_2, \dots, x_n)$. С каждой диетой связывается соответствующее ей количество питательных веществ и микро-

элементов — белков, жиров, углеводов, клетчатки, железа, кальция, витаминов и других жизненно необходимых компонент. Пусть величина a_{ij} характеризует содержание i -го питательного вещества в единице j -го продукта ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$). Тогда общее содержание i -го питательного вещества в принятой диете равно

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n.$$

Обозначим минимальную ежесуточную потребность организма в i -м питательном веществе через b_i . Это означает, что диета должна удовлетворять ограничениям

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \quad (i = 1, 2, \dots, m). \quad (2.1)$$

С другой стороны, суточное потребление каждого продукта ограничено запасами, а также возможностями и сроками их пополнения. Отсюда дополнительные условия на компоненты диеты

$$0 \leq x_j \leq a_j \quad (j = 1, 2, \dots, n), \quad (2.2)$$

где величины a_j определяются наличными ресурсами продуктов P_j и временем, на которое они рассчитаны. Если c_j — стоимость единицы j -го продукта, то стоимость всей диеты определяется линейной формой

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n. \quad (2.3)$$

Оптимальной диетой следует признать такую систему ежесуточного потребления продуктов $X = \{x_1, x_2, \dots, x_n\}$, которая обращает в минимум линейную форму (2.3) при соблюдении условий (2.1) и (2.2).

В ряде практических задач на составление диеты необходимо также учитывать ограничения сверху на потребление некоторых питательных веществ, т. е. вместо условий (2.1) следует рассматривать ограничения вида

$$\bar{b}_i \geq a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \quad (i = 1, 2, \dots, m).$$

При использовании нормативных расчетов для определения рациональной структуры пищевой промышленности необходимо учитывать традиции в потреблении, вкусы и привычки населения. Первый вариант норм потребления, разработанный Вычислительным центром Госэкономсовета совместно с Институтом питания Академии медицинских наук, основывался на модели, близкой

к схеме (2.1) — (2.3). В результате расчетов оказалось, что наиболее экономичная диета, удовлетворяющая физиологическим требованиям, не включает ряд традиционных для населения продуктов, таких, как ржаной хлеб, капуста, говядина и крупа, и содержит другие продукты, например треску, в гораздо большем объеме, чем это определяется вкусами и привычками населения.

Государство может в известных пределах влиять не только на структуру производства продуктов питания, но и на спрос. Это может быть достигнуто путем снижения цен на продукты, потребление которых должно быть стимулировано, и сохранения повышенных цен на товары, потребление которых следует ограничить. Указанное обстоятельство не отменяет тем не менее проблемы учета традиций, вкусов и привычек.

Простейший в принципе, но достаточно трудоемкий в реализации способ учета перечисленных факторов заключается в следующем. Вводятся дополнительные условия, ограничивающие сверху потребление продуктов, которые содержатся в предварительном плане в избытке, и снизу — потребление традиционных для населения продуктов, не включенных в первый вариант плана. Обычно таких дополнительных параметров-границ немного. Решая ряд задач линейного программирования для различных рационально выбранных значений границ или используя методы так называемого параметрического программирования, вычисляют значение стоимости оптимальной диеты как функцию от введенных параметров-границ. Задавая затем допустимой величиной превышения стоимости диеты над ее минимальной стоимостью, определяют возможные коррективы плана на сложившиеся традиции в потреблении.

При разработке диеты для специализированных больниц вряд ли можно принять в качестве целевой функции решения задачи стоимость питания. Показатель качества диеты определяется особенностями заболеваний. Стоимость диеты в этом случае — ограничение, определяемое бюджетом органов здравоохранения. Ряд заболеваний полностью исключает из потребления отдельные виды продуктов, например сахар, жиры и др., или существенно ограничивает их содержание в пище. Все эти условия могут

быть учтены при составлении модели задачи в виде линейных ограничений.

В последующих параграфах мы увидим, что к схемам различных вариантов нормирования потребления может быть сведено большое число задач, связанных с составлением оптимальных рецептов, сплавов и смесей, удовлетворяющих заранее заданным требованиям.

§ 3. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К ЭНЕРГЕТИКЕ

В настоящее время разработано и освоено значительное число проектов электростанций, использующих различные источники энергии. Имеется ряд стандартных проектов гидроэлектростанций и тепловых станций, применяющих различные виды горючего, — уголь, нефть, торф, горючие сланцы и др. Освоено несколько типов атомных электростанций. Разработаны приливные электростанции, использующие энергию морского прилива. Рассматриваются и другие принципы получения электрической энергии (например, преобразование солнечной энергии, энергии ветра, энергии вулканической деятельности земли и т. д.).

При решении вопроса об электрификации района, располагающего различными источниками энергии, важно выбрать рациональный набор типов станций, способных удовлетворить потребности района при минимальных затратах. Любой набор станций, составляющих энергетическую систему района, должен (см. [23]) обеспечить заданные значения гарантированной мощности (A), пиковой мощности (B) и заданный уровень годовой выработки энергии (C). При этом затраты на капитальное строительство не должны превысить фиксированной суммы (D). Величины A , B , C и D определяются перспективным планом развития экономики района.

Пусть природные условия района позволяют рассматривать возможность строительства в районе любого из n типов электростанций. Обозначим через x_j число станций j -го типа. Введем, кроме того, следующие обозначения: a_j — гарантированная мощность, обеспечиваемая одной станцией j -го типа; b_j — пиковая мощность одной станции

j -го типа; c_j — годовая выработка энергии одной станцией j -го типа; d_j — затраты на строительство одной станции j -го типа; f_j — суммарные годовые эксплуатационные расходы (обслуживание и управление) на одну станцию j -го типа.

Оптимальная в смысле минимума эксплуатационных расходов энергосистема состоит из x_j станций j -го типа ($j = 1, 2, \dots, n$). Числа x_j обеспечивают минимум линейной формы

$$L = \sum_{j=1}^n f_j x_j \quad (3.1)$$

при условиях

$$\sum_{j=1}^n a_j x_j \geq A, \quad (3.2)$$

$$\sum_{j=1}^n b_j x_j \geq B, \quad (3.3)$$

$$\sum_{j=1}^n c_j x_j \geq C, \quad (3.4)$$

$$\sum_{j=1}^n d_j x_j \leq D, \quad (3.5)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (3.6)$$

Показатель качества (3.1) решения задачи равен суммарным годовым расходам всех станций энергосистемы на обслуживание, управление и уход за оборудованием. Ограничения (3.2) — (3.4) обеспечивают заданный уровень гарантированной и пиковой мощностей системы и требуемую годовую выработку энергии по району. Ограничение (3.5) гарантирует, что суммарные затраты на строительство станций энергосистемы не превысят размеров запланированных капиталовложений. Естественные ограничения (3.6) определяются физическим смыслом переменных x_j . Заметим, кроме того, что по своему физическому смыслу переменные x_j — целые числа. Решение задачи (3.1) — (3.6) может привести к дробным значениям компонент x_j . Округление значений x_j до ближайших целых чисел, как правило, не приводит к существенным отклонениям от

оптимального плана, если $x_j \gg 1$. При малых значениях компонент оптимального плана задачи (3.1) — (3.6) округление составляющих x_j решения недопустимо. Полученный таким образом план может оказаться существенно менее эффективным, чем оптимальный целочисленный план задачи. Может также оказаться, что округление компонент оптимального плана до ближайших целых чисел приводит к нарушению ограничений задачи. В таких случаях следует привлечь для решения задачи методы так называемого *целочисленного программирования* (см., например, [12], гл. 5).

Задача (3.1) — (3.6) содержит четыре условия общего вида ($m=4$). Отсюда следует*) (это будет обосновано в § 3 гл. 3), что оптимальная энергосистема, определяемая решением задачи (3.1) — (3.6), будет содержать не более четырех типов станций. Необходимость учета дополнительных ограничений заставляет, вообще говоря, увеличить число типов станций, входящих в состав оптимальной энергосистемы. Дополнительные условия могут определяться, например, ограниченным количеством того или иного оборудования, необходимого всем или некоторым станциям системы. Лимитирующим фактором при составлении плана развития энергетики района может оказаться также ограниченная возможность обеспечения района квалифицированными инженерно-техническими кадрами. Все подобные ограничения записываются так же, как и ограничения (3.5).

Строительство станций по новым проектам требует создания нового оборудования. Развертывание нового производства экономически оправдано лишь в тех случаях, когда потребность в его продукции превышает некоторый минимум. Отсюда новые ограничения вида

$$\sum_{j=1}^n e_j^{(i)} x_j \geq \varepsilon_{(i)}, \quad i = 1, 2, \dots, s,$$

где $e_j^{(i)}$ — потребность каждой станции j -го типа в i -м виде нового оборудования, а $\varepsilon_{(i)}$ — минимальный объем продукции

*) В предположении целочисленности оптимального плана задачи (3.1) — (3.6).

i -го вида, оправдывающий развертывание нового производства.

Отдельные типы станций, нерентабельные с точки зрения современных технических возможностей, являются тем не менее перспективными предприятиями, и накопление опыта их строительства и эксплуатации — необходимое условие технического прогресса. С другой стороны, имеются определенные проекты станций, экономичных в настоящее время, но по тем или иным причинам неперспективных (например, из-за ограниченных ресурсов соответствующего источника энергии). Учет последних замечаний приводит к замене ограничений вида (3.6) условиями вида

$$\alpha_j \leq x_j \leq \beta_j, \quad j = 1, 2, \dots, n,$$

где α_j и β_j предполагаются заданными величинами. Значения α_j и β_j определяются при планировании на более высоком уровне, исходящем из более далекой перспективы.

§ 4. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К МЕТАЛЛУРГИИ

В литературе описаны модели разных процессов металлургического производства, для исследования которых успешно использованы методы линейного программирования [39, 41 и 8]. В частности, к схемам линейного программирования сводятся задачи составления экономной шихты для выплавки чугуна в доменных печах и стали в мартеновских и электросталеплавильных печах. К общим и специальным моделям линейного программирования могут быть сведены задачи оптимального раскрытия различных видов проката и рационального разделения слитков на объемные заготовки заданной формы. Методами линейного программирования решались также задачи распределения проката различных видов стального листа между станами. Рассмотрим некоторые из перечисленных моделей, сводящихся к задачам линейного программирования общего вида.

4.1. Подбор шихты для выплавки чугуна. Задача выбора рационального состава шихты относится к классу

задач линейного программирования, известных под названием «задачи о смеси». Однако постановка этой задачи имеет некоторые особенности, отличающие ее от модели задачи о диете и от схемы получения авиационных бензинов с заданными свойствами. Дело в том, что под оптимальной шихтой вряд ли целесообразно понимать наиболее дешевую смесь шихтовых материалов, химический состав которой удовлетворяет определенным свойствам. Шихта является исходным продуктом для выплавки металла. Различный состав шихты определяет различный технологический процесс плавки. Поэтому естественно выбрать показателем качества шихты стоимость выплавленного металла.

В модели, предложенной в работе [39], процесс производства металла рассматривается не как выплавка его из смеси разных шихтовых материалов, а как смешивание металлов разного химического состава, каждый из которых как бы выплавлен из одного вида шихтового материала.

В состав шихтовых материалов входят руды разных сортов и марок, окалина, мартеновский шлак, скрап и т. д. Различные шихтовые материалы определяют разный химический состав выплавленного чугуна. В качественном чугуне процент серы, марганца, фосфора и некоторых других элементов не должен превышать заданных величин. Ряд компонент шихты представляет собой отходы других цехов металлургического производства (например, окалина и мартеновский шлак). Ресурсы этих составляющих ограничены производством соответствующих цехов. Имеются и технологические ограничения на доли определенных шихтовых материалов в завалке.

Оптимальной шихтой будем считать такой состав шихтовых материалов, который обеспечивает при минимальных затратах выплавку чугуна, удовлетворяющего требуемым условиям по химическому составу. При этом должны быть также учтены ограничения по отдельным компонентам шихты.

Переведем задачу на математический язык. В соответствии с принятой моделью процесса производства металла введем переменные x_j , обозначающие долю гипотетического чугуна, выплавленного из j -го шихтового

материала. Пусть общее число сортов шихтового материала равно n . Ясно, что

$$\sum_{j=1}^n x_j = 1. \quad (4.1)$$

Введем, кроме того, следующие обозначения: a_{ij} — процент i -го элемента (серы, марганца, фосфора и др.) в чугуна, который гипотетически мог бы быть получен из j -го шихтового материала; a_i — максимально допустимый процент i -го элемента в чугуна; d_j — максимальная доля чугуна, которая могла бы быть выплавлена из j -го шихтового материала; c_j — издержки на 1 т чугуна, который гипотетически мог бы быть получен из j -го шихтового материала.

Ограничения по химическому составу чугуна записываются в виде

$$\sum_{j=1}^n a_{ij} x_j \leq a_i, \quad i = 1, 2, \dots, r, \quad (4.2)$$

где r — число элементов, наличие которых в качественном чугуна должно быть строго ограничено.

Технологические условия и ограниченные ресурсы отдельных составляющих шихты приводят к неравенствам

$$0 \leq x_j \leq d_j, \quad j = 1, 2, \dots, n. \quad (4.3)$$

Суммарные издержки на производство 1 т смеси различных видов чугуна (гипотетически произведенных из разных шихтовых материалов) могут быть записаны в виде линейной формы

$$L = \sum_{j=1}^n c_j x_j. \quad (4.4)$$

Задача сводится, таким образом, к вычислению x_j , обращающих в минимум линейную форму (4.4) при условиях (4.1)—(4.3). Оптимальные пропорции между отдельными компонентами шихты определяются составляющими x_j^* решения задачи (4.1)—(4.4). Коэффициент пропорциональности определяется нормами расхода шихтовых материалов на 1 т чугуна.

4.2. Подбор шихты для выплавки легированной стали.

Определение оптимального состава шихты для выплавки легированной стали в электропечах — гораздо более сложная задача, чем подбор шихты для доменной выплавки чугуна. Выбор состава шихты осложняется в этом случае рядом обстоятельств. Во-первых, при выплавке стали шихтовые материалы вводятся в печь не сразу, а по фазам процесса. Различные варианты распределения составляющих шихты по фазам существенно сказываются на себестоимости слитков. Во-вторых, расход шихтовых материалов в значительной мере определяется различными параметрами технологического процесса. К таким параметрам относятся объем кислорода, вдуваемого в печь, и скорость вдувания, температура ванны в начале и в конце периода продувки ее кислородом, состав шлака и т. п. Подбор рационального состава шихты сводится, таким образом, к выбору оптимальной технологии плавки. Это весьма сложная задача.

В работе [41] приводятся следующие предложения, существенно упрощающие постановку задачи:

1°. Заранее отбирается несколько вариантов технологии плавки, которые отличаются по основным параметрам процесса, влияющим на состав шихты.

2°. Число фаз процесса, требующих ввода шихтовых материалов, сводится к трем — основная садка, восстановление и доводка.

3°. Для каждого варианта технологии методами линейного программирования рассчитывается оптимальная шихта.

4°. Окончательный выбор варианта и, следовательно, состава шихты по фазам производится на основе дополнительного экономического анализа.

Рассмотрим модель выбора шихты для фиксированного варианта технологии.

Так же как и в предыдущей задаче, процесс производства металла рассматривается не как выплавка стали из смеси разных шихтовых материалов, а как смешивание металлов, выплавленных из отдельных составляющих шихты на отдельных фазах технологического процесса.

Введем следующие обозначения: x_{ij} — доля выплавленной стали, приходящаяся на j -ю компоненту шихты

($j = 1, 2, \dots, n$), введенную в печь на i -й фазе процесса ($i = 1, 2, 3$); $a_{ij}^{(s)}$ — процент s -го элемента (хрома, кремния и др.), содержащийся в стали, которая отвечает j -й составляющей шихты и i -й фазе процесса ($s = 1, 2, \dots, r$); $\underline{a}_i^{(s)}$ и $\bar{a}_i^{(s)}$ — нижний и верхний допуски на процентное содержание s -го элемента в i -й фазе; $a^{(s)}$ — максимально допустимый процент s -го элемента в стали; d_j — определяемая технологическими требованиями и ресурсами максимальная доля стали, приходящаяся на j -ю компоненту шихты; c_{ij} — издержки на 1 т легированной стали, которая гипотетически могла бы быть получена из j -й компоненты шихты, введенной в печь на i -й фазе процесса.

По определению искоемых переменных

$$\sum_{i=1}^3 \sum_{j=1}^n x_{ij} = 1. \quad (4.5)$$

Технологические условия, связанные с химическим составом плавки на отдельных ее фазах, записываются в виде

$$\underline{a}_i^{(s)} \leq \sum_{j=1}^n a_{ij}^{(s)} x_{ij} \leq \bar{a}_i^{(s)}, \quad i = 1, 2, 3; \quad s = 1, 2, \dots, r. \quad (4.6)$$

Ограничения по химическому составу легированной стали представляют собой неравенства вида

$$\sum_{i=1}^3 \sum_{j=1}^n a_{ij}^{(s)} x_{ij} \leq a^{(s)}, \quad s = 1, 2, \dots, r. \quad (4.7)$$

Ограниченные ресурсы по отдельным шихтовым материалам и технологические требования приводят к дополнительным условиям

$$\sum_{i=1}^3 x_{ij} \leq d_j, \quad j = 1, 2, \dots, n. \quad (4.8)$$

Кроме того, должны соблюдаться естественные ограничения

$$x_{ij} \geq 0, \quad i = 1, 2, 3; \quad j = 1, 2, \dots, n. \quad (4.9)$$

Суммарные затраты на производство 1 т легированной стали, которая получена в соответствии с принятой

моделью, как смесь из металла, выплавленного из отдельных сортов шихтовых материалов на разных фазах процесса, выражаются суммой

$$L = \sum_{i=1}^3 \sum_{j=1}^n c_{ij} x_{ij}. \quad (4.10)$$

Задача сводится к вычислению параметров x_{ij} , на которых достигается минимум линейной формы (4.10) при условиях (4.5)—(4.9).

Рациональные объемы различных составляющих шихты, вводимых в печь на той или иной фазе технологического процесса, пропорциональны соответствующим компонентам x_{ij}^* решения задачи (4.5)—(4.10). Коэффициент пропорциональности определяется нормами расхода шихты на 1 т легированной стали.

Задача (4.5)—(4.10) содержит определенные особенности по сравнению с задачей линейного программирования общего вида. Тем не менее эта задача рассматривается в настоящей главе среди задач, решаемых общими методами линейного программирования, поскольку ее специфика не дает основания рассчитывать на специальные методы, существенно упрощающие решение.

Известны работы по применению линейного программирования к рациональному раскрою холоднокатаной стальной ленты и раскрою листовой стали. Применение листовой стали в народном хозяйстве исчисляется миллионами тонн, а расходы при ее раскросе очень значительны. Формализация задач, связанных с рациональным раскросом материалов, выделена в отдельный параграф (см. § 9).

§ 5. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К НЕФТЯНОЙ ИНДУСТРИИ

Линейное программирование нашло широкое применение при составлении проектов и анализе разработки нефтяных месторождений. По этим вопросам в нашей и иностранной периодической литературе опубликовано значительное количество статей (см., например, [24, 32, 30 и 45]). Разнообразные задачи линейного программи-

рования возникают также в нефтепереработке и, в частности, в задачах смешивания нефтепродуктов для получения авиационных бензинов по заданным техническим условиям [27, 37 и 42].

Рассмотрим некоторые из приложений линейного программирования к задачам нефтяной индустрии.

5.1. Один из эффективных методов разработки нефтяных месторождений — это метод обводнения, при котором накачивание воды обеспечивает условия фонтанирования скважин. Однако независимо от технологических способов использования обводнения всегда наступает момент, когда накачиваемая вода начинает прорываться к эксплуатационным скважинам. Нефть, полученная из таких скважин, требует последующей обработки на обезвоживающих установках. При этом, естественно, повышается ее себестоимость. Возникает необходимость в обеспечении отбора нефти между скважинами так, чтобы обеспечить выполнение планового задания по добыче при минимальной себестоимости. При этом должны быть удовлетворены технологические ограничения, обусловленные физическими свойствами жидкостей, насыщающих пласт, и предполагаемым способом разработки. Тем не менее значительный интерес представляют частные постановки, сводящиеся к задаче линейного программирования. Больше того, задача в общей постановке может быть сведена к серии задач линейного программирования или к так называемой задаче параметрического программирования (см., например, [12], гл. 3).

Введем понятия, необходимые для построения математической модели разработки обводненных месторождений.

Пусть $x_j^{(в)}$ и $x_j^{(н)}$ — соответственно количество воды и нефти, содержащейся в суммарном дебите j -й скважины $x_j = x_j^{(в)} + x_j^{(н)}$. Коэффициентом обводнения называется отношение $\alpha_j = x_j^{(в)} / x_j$. Каждая скважина характеризуется своим коэффициентом обводнения α_j , который мы в дальнейшем будем считать заданной величиной.

Для нормального функционирования скважин необходимо, чтобы:

1) забойные давления p_i на каждую скважину были не ниже некоторого фиксированного давления p_i^* , называемого давлением насыщения или давлением разгазирования;

2) динамические давления q_i , необходимые для фонтанирования скважин, должны превышать некоторые фиксированные величины q_i^* .

При движении жидкости, подчиняющейся линейному закону фильтрации, и водонапорном режиме пласта величины p_i и q_i являются линейными функциями дебитов скважин x_j :

$$p_i = \sum_{j=1}^n a_{ij} x_j, \quad q_i = \sum_{j=1}^n b_{ij} x_j, \quad i = 1, 2, \dots, n, \quad (5.1)$$

где n — число скважин, а постоянные a_{ij} и b_{ij} зависят от конкретных нефтепромысловых условий эксплуатации месторождения (от взаимного расположения скважин, характеристик пласта и метода разработки).

Суммарная добыча обезвоженной нефти определяется выражением

$$Q = \sum_{j=1}^n x_j^{(H)} = \sum_{j=1}^n (x_j - x_j^{(B)}) = \sum_{j=1}^n (1 - \alpha_j) x_j. \quad (5.2)$$

Затраты на эксплуатацию обводненных месторождений можно разделить на две составляющие. Первая составляющая не зависит от дебита добычи. Это затраты на оборудование, отчисления на зарплату и капитальный ремонт, а также внутрипромысловые расходы. Обозначим суммарные расходы первой группы через S_1 :

$$S_1 = \sum_{j=1}^n s_{1j},$$

где s_{1j} — расходы первой группы по j -й скважине.

Расходы второй группы линейно зависят как от суммарного дебита добычи, так и от количества воды и нефти в суммарном дебите каждой скважины:

$$S_2 = \sum_{j=1}^n s_{2j} = \sum_{j=1}^n (a_j x_j + b_j x_j^{(B)} + c_j x_j^{(H)}),$$

где a_j — расходы на поддержание пластового давления; b_j — расходы на сброс загрязненных вод; c_j — расходы на переработку и перекачку нефти из j -й скважины. Все расходы второй группы задаются на 1 м³ добываемой жидкости. Затраты на обезвоживание нефти линейно зависят от величин $x_j^{(в)}$ и $x_j^{(н)}$. Поэтому их можно учесть вместе с эксплуатационными расходами, изменив соответствующим образом коэффициенты b_j и c_j .

Выражение для суммарных затрат можно записать в следующем виде:

$$S = S_1 + S_2 = \sum_{j=1}^n \{ s_{1j} + [a_j + b_j a_j + c_j (1 - \alpha_j)] x_j \},$$

или

$$S = \sum_{j=1}^n (s_{1j} + r_j x_j),$$

где $r_j = a_j + b_j a_j + c_j (1 - \alpha_j)$.

Себестоимость 1 м³ нефти равна

$$M = \frac{S}{Q} = \frac{\sum_{j=1}^n (s_{1j} + r_j x_j)}{\sum_{j=1}^n (1 - \alpha_j) x_j}. \quad (5.3)$$

Вычисление дебитов добычи каждой скважины (величин $x_j \geq 0$), минимизирующих себестоимость (5.3) нефти при условии выполнения планового задания ($Q \geq Q^*$) и удовлетворения технологических требований ($p_i \geq p_i^*$, $q_i \geq q_i^*$, $i = 1, 2, \dots, n$), — задача так называемого *дробно-линейного программирования*. Ее можно свести к задаче линейного программирования. Однако мы на этом вопросе останавливаться не будем.

В ряде случаев представляется целесообразным распределять отбор нефти между скважинами, исходя из стремления обеспечить по крайней мере заданный объем обезвоженной нефти при минимальных затратах. При этом, естественно, должно быть гарантировано выполнение технологических условий. Мы пришли, таким образом, к задаче линейного программирования.

Требуется обратить в минимум линейную форму

$$S_2 = S - S_1 = \sum_{j=1}^n r_j x_j \quad (5.4)$$

при условиях

$$\sum_{j=1}^n (1 - \alpha_j) x_j \geq Q^*, \quad (5.5)$$

$$\sum_{j=1}^n a_{ij} x_j \geq p_i^*, \quad i = 1, 2, \dots, n, \quad (5.6)$$

$$\sum_{j=1}^n b_{ij} x_j \geq q_i^*, \quad i = 1, 2, \dots, n, \quad (5.7)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (5.8)$$

Если допустимые затраты на эксплуатацию месторождения и обработку нефти заранее заданы, целесообразно изменить постановку задачи. В этом случае естественно считать выбор дебитов добычи отдельных скважин лучшим, если он обеспечивает максимальный объем производства обезвоженной нефти при допустимых затратах и соблюдении всех технологических требований. Задача сводится к вычислению чисел x_j , обращающих в максимум линейную форму

$$Q = \sum_{j=1}^n (1 - \alpha_j) x_j \quad (5.9)$$

при условиях

$$\sum_{j=1}^n r_j x_j \leq S_2, \quad (5.10)$$

$$\sum_{j=1}^n a_{ij} x_j \geq p_i^*, \quad i = 1, 2, \dots, n, \quad (5.11)$$

$$\sum_{j=1}^n b_{ij} x_j \geq q_i^*, \quad i = 1, 2, \dots, n, \quad (5.12)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (5.13)$$

Перечисленные выше задачи допускают различные модификации, анализ которых также может быть проведен методами линейного программирования. Не во всех

приложениях можно считать коэффициенты обводнения α_j заданными величинами. Часто определению подлежат как отбор нефти, так и закачка воды. Показателями качества разработки месторождения могут служить величины отбора нефти или закачки воды, или, наконец, эксплуатационные расходы на разработку месторождения. Постановка задачи определяет способ использования пластовой энергии. Выбор постановки задачи в свою очередь обусловлен конкретными особенностями разрабатываемого месторождения.

В литературе имеются указания на использование методов линейного программирования для управления движением контура нефтеносности и других проектных и эксплуатационных задач по разработке нефтеносных пластов в различных режимах.

5.2. Другая область приложения линейного программирования к нефтяной индустрии — это задачи переработки нефтепродуктов. Задача оптимального смешивания нефтепродуктов для получения авиационного бензина с заданными техническими характеристиками является специальной задачей линейного программирования. Ниже излагается более простая модель переработки сырой нефти, анализ которой сводится к общей задаче линейного программирования [37].

На нефтеперегонный завод ежемесячно поступает n сортов сырой нефти в определенных количествах (a_j тонн j -го сорта). Завод должен изготовить из сырой нефти m видов нефтепродуктов (жидкие топлива и смазочные масла). Некоторые нефтепродукты могут быть получены при перегонке любого сорта нефти, но в различных пропорциях. Другие продукты получаются только при перегонке отдельных видов нефти. Заданы характеристики сортов нефти по выходам и по рентабельности: a_{ij} — норма выхода i -го продукта из 1 m нефти j -го сорта; c_j — затраты на перегонку 1 m нефти j -го сорта.

Пусть предусмотренное планом задание на производство i -го вида нефтепродукта равно b_i тонн.

Необходимо определить, какое количество нефти каждого сорта следует переработать для выполнения плана по ассортименту при минимальных затратах. Обозначим через x_j количество перерабатываемой сырой

нефти j -го сорта. В принятых обозначениях задача сводится к вычислению значений переменных x_j ($j=1, 2, \dots, n$), обращающих в минимум линейную форму

$$L = \sum_{j=1}^n c_j x_j \quad (5.14)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i=1, 2, \dots, m, \quad (5.15)$$

$$0 \leq x_j \leq a_j, \quad j=1, 2, \dots, n. \quad (5.16)$$

Линейная форма (5.14) соответствует суммарным затратам на перегонку нефти. Условия (5.15) гарантируют выполнение плана по каждому виду нефтепродуктов. Условия (5.16) выражают ограничения по количеству поступающей нефти каждого сорта.

Задача (5.14)—(5.16) представляет собой задачу линейного программирования общего вида с двусторонними ограничениями.

Может оказаться, что емкости для хранения непереработанной сырой нефти j -го сорта и изготовленного нефтепродукта i -го типа ограничены соответственно величинами \bar{a}_j и \bar{b}_i . В этом случае условия (5.15) и (5.16) заменяются соответственно на следующие:

$$b_i \leq \sum_{j=1}^n a_{ij} x_j \leq \bar{b}_i, \quad i=1, 2, \dots, m, \quad (5.17)$$

$$0 \leq x_j \leq \alpha_j, \quad j=1, 2, \dots, n, \quad (5.18)$$

где $\alpha_j = \min(a_j, \bar{a}_j)$.

§ 6. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ В СЕЛЬСКОМ ХОЗЯЙСТВЕ

Целый ряд задач сельскохозяйственной практики приводится к моделям линейного программирования. Сюда относятся задачи о рациональном распределении посевных площадей, о составлении экономного рациона от-

корма скота, о специализации сельскохозяйственного производства, о графике работы сельскохозяйственных машин и т. д. Здесь будут описаны лишь модели, сводящиеся к общей задаче линейного программирования.

6.1. Распределение посевной площади на целине. Во вновь осваиваемых районах, особенно на целине, равномерная загрузка рабочей силы в течение года является весьма важным фактором. Часто этот фактор оказывается решающим для закрепления населения за определенными районами. Проводимые обычно мероприятия, такие, как перенос отпусков на зиму, строительство ремонтных мастерских, консервных заводов и т. д., не решают полностью проблему равномерной загрузки основной массы рабочей силы. Поэтому при планировании сельскохозяйственных работ в совхозах необходимо известную часть посевной площади отводить под культуры, главным образом технические, требующие затраты труда в зимнее время года.

Пусть требуется распределить площадь под n культур. Разобьем год на m периодов не обязательно одинаковой продолжительности. Естественно, например, май считать за три периода, а всю зиму — за один.

Введем следующие обозначения: x_j — площадь, отведенная под j -ю культуру; b_i — количество рабочей силы, которой располагает совхоз в i -й период; d_i — число людей, которое может быть в i -й период занято не сельскохозяйственными работами; a_{ij} — число людей, которое следует занять в i -й период на j -й культуре, если под нее отведена единица посевной площади; c_j — прибыль от j -й культуры, приходящаяся на единицу посевной площади.

Общие доходы совхоза равны

$$L = \sum_{j=1}^n c_j x_j. \quad (6.1)$$

Наивыгоднейшее распределение посевной площади определяется здесь системой чисел (x_1, x_2, \dots, x_n) , обращающих линейную форму (6.1) в максимум при условиях

$$x_j \geq \frac{a_j}{\alpha_j}, \quad j = 1, 2, \dots, n, \quad (6.2)$$

где a_j — плановое задание по производству j -й культуры (некоторые a_j могут быть равны нулю), α_j — урожайность j -й культуры;

$$b_i - d_i \leq \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m; \quad (6.3)$$

условия (6.3) гарантируют равномерную загрузку рабочей силы в течение года;

$$\sum_{j=1}^n x_j \leq a, \quad (6.4)$$

где a — посевная площадь совхоза.

Мы пришли к задаче линейного программирования общего вида, в которой требуется определить параметры x_j ($j = 1, 2, \dots, n$), обращающие в максимум линейную форму (6.1) при условиях (6.2) — (6.4).

6.2. Выбор рациона откорма скота. Выбор наиболее экономного набора продуктов, содержащих необходимое количество питательных веществ, является одной из разновидностей задачи о смеси. Особенности задачи заключаются в учете ограничений, вытекающих из непосредственных условий сельскохозяйственного производства.

В простейшей постановке задача о выборе оптимального рациона откорма скота может быть сформулирована следующим образом (см. [43]). Пусть хозяйство располагает n видами кормов (сено, силос, свекла, концентраты, картофель и т. д.). Предполагаются известными следующие характеристики корма. Одна тонна j -го вида корма ($j = 1, 2, \dots, n$) содержит a_{0j} кормовых единиц. При этом питательная норма на стадо хозяйства равна a_0 кормовых единиц. Каждый из видов корма содержит некоторые из питательных веществ (белки, кальций, фосфор, витамины и др.), которые должны быть включены в рацион. Обозначим содержание i -го вида питательного вещества ($i = 1, 2, \dots, m$) в 1 т j -го вида корма через a_{ij} кг. Для откорма стада требуется не менее a_i кг i -го питательного вещества.

Чтобы разнообразить рацион, целесообразно иметь не менее \underline{a}_j тонн j -го вида корма. С другой стороны, запасы

j -го вида корма на планируемый период ограничены величиной \bar{d}_j . Наконец, зоотехнические нормы требуют определенного соотношения между объемом сочных и грубых кормов. Пусть, например, объем сочных кормов должен быть в рационе в λ раз больше объема грубых кормов.

Обозначим через x_j ($j = 1, 2, \dots, n$) количество тонн j -го вида корма на планируемый промежуток времени. Введем два множества индексов I_1 и I_2 . Индекс j будем относить к множеству I_1 , если j -й вид корма сочный, и к множеству I_2 , если этот индекс соответствует грубому корму. Обозначим через c_j себестоимость производства одной тонны кормов j -го вида.

В принятых обозначениях задача составления наиболее экономного рациона откорма скота сводится к задаче линейного программирования, в которой требуется вычислить переменные x_j , обеспечивающие минимум линейной формы:

$$L = \sum_{j=1}^n c_j x_j \quad (6.5)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq a_i, \quad i = 0, 1, 2, \dots, m, \quad (6.6)$$

$$\sum_{j \in I_1} x_j \geq \lambda \sum_{j \in I_2} x_j, \quad (6.7)$$

$$\underline{d}_j \leq x_j \leq \bar{d}_j, \quad j = 1, 2, \dots, n. \quad (6.8)$$

Показатель качества (6.5) рациона — суммарная стоимость производства кормов. Условия (6.6) обеспечивают требуемые питательные вещества в нужных количествах. При $i = 0$ условие (6.6) гарантирует необходимое количество кормовых единиц. Ограничение (6.7) фиксирует необходимое соотношение между сочными и грубыми кормами. И, наконец, условия (6.8) определяют наличный запас кормов и требования разнообразия рациона.

6.3. Разработка рациональной структуры отраслей животноводства. В работе [43] поставлена задача об установлении оптимальной численности каждого вида скота, исходя из имеющихся ресурсов.

Возможности животноводства ограничиваются главным образом кормовой базой. В последующих рассуждениях предполагается, что экономный рацион откорма разных видов скота уже определен (например, в соответствии с указаниями предыдущего пункта).

В зависимости от того, какой смысл вкладывать в понятие «оптимальная структура животноводства», возможны различные постановки задачи. Принимая в качестве критерия оптимальности структуры стоимостной критерий, можно вычислить количество голов скота каждого вида, при котором имеющийся корм гарантирует хозяйству максимальный доход от продукции животноводства.

Формализация задачи требует следующих обозначений: x_j — число голов скота j -го вида ($j = 1, 2, \dots, n$); p_j — доход, получаемый хозяйством от одной головы скота j -го вида; b_{kj} — нормы потребления корма k -го сорта в расчете на одну голову скота j -го вида ($k = 1, 2, \dots, s$); b_k — ресурсы корма k -го сорта.

Задача сводится к вычислению величин x_j , для которых линейная форма

$$L = \sum_{j=1}^n p_j x_j \quad (6.9)$$

достигает максимального значения при условиях

$$\sum_{j=1}^n b_{kj} x_j \leq b_k, \quad k = 1, 2, \dots, s, \quad (6.10)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (6.11)$$

Ограничения по различным производственным факторам (например, по рабочей силе или сельскохозяйственной технике), если они существенны, учитываются условиями вида (6.10).

В ряде случаев более естественным представляется натуральный показатель качества структуры отраслей животноводства. И в этом случае возможны различные постановки задачи. Так, например, может быть задан рациональный ассортимент — требуемое соотношение между различными видами продукции животноводства (мясо, молоко, шерсть и т. д.). Структура отраслей животноводства

водства определяется тогда, исходя из стремления обеспечить максимальный приведенный объем продукции животноводства. При этом должны соблюдаться требуемая ассортиментность и учитываться возможности кормовой базы. Математически выбор структуры сводится к вычислению чисел $x_1, x_2, \dots, x_n, x_{n+1}$, для которых достигается максимум величины

$$L = x_{n+1} \quad (6.12)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq \mu_i x_{n+1}, \quad i = 1, 2, \dots, m, \quad (6.13)$$

$$\sum_{j=1}^n b_{kj} x_j \leq b_k, \quad k = 1, 2, \dots, s, \quad (6.14)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (6.15)$$

Здесь переменные x_j ($j = 1, 2, \dots, n$) и величины b_{kj} и b_k сохраняют тот же смысл, что и в задаче (6.9) — (6.11). Под a_{ij} подразумевается объем продукции животноводства i -го вида (мясо, молоко, шерсть и т. д.) от одной головы скота j -й группы. Отношения $\mu_1 : \mu_2 : \dots : \mu_m$ определяют требуемую ассортиментность различных видов продукции животноводства. Таким образом, объем продукции животноводства характеризуется величиной x_{n+1} .

Приведем еще одну постановку задачи об определении структуры отраслей животноводства, которая в ряде случаев представляется более целесообразной.

Обычно в каждом животноводческом хозяйстве одна из отраслей животноводства является профилирующей. Качество структуры определяется по объему производства профилирующей продукции, а по остальным продуктам задается план, выполнение которого должно быть гарантировано выбором структуры. При этом, естественно, учитывается обеспеченность животноводства кормами.

Сохраним за обозначениями x_j, a_{ij}, b_{kj}, b_k тот же смысл, что и в задачах (6.9) — (6.11) и (6.12) — (6.15). Введем, кроме того, следующие дополнительные обозначения: c_j — объем продукции профилирующего вида в расчете на одну голову скота j -й группы; a_i — плановое задание по i -му виду продукции животноводства.

В принятых обозначениях задача сводится к определению чисел $x_j \geq 0$ ($j=1, 2, \dots, n$) (количества голов скота каждой группы), обеспечивающих максимальный объем профилирующей продукции

$$L = \sum_{j=1}^n c_j x_j \quad (6.16)$$

при выполнении плана по другим продуктам животноводства

$$\sum_{j=1}^n a_{ij} x_j \geq a_i, \quad i=1, 2, \dots, m, \quad (6.17)$$

и учете возможностей кормовой базы

$$\sum_{j=1}^n b_{kj} x_j \leq b_k, \quad k=1, 2, \dots, s. \quad (6.18)$$

Мы снова пришли к задаче линейного программирования общего вида.

6.4. Структура посевов кормовых культур. В тех случаях, когда требования к животноводческой продукции уже зафиксированы, возникает задача планирования кормовой базы. Требуется распределить отведенную площадь под отдельные кормовые культуры так, чтобы при минимальных затратах удовлетворить животноводческое хозяйство необходимыми кормами. При этом обычно приходится учитывать также ограничения по трудовым ресурсам, по тракторному парку, по горючему, по удобрениям, по орошению и другим факторам, обеспечивающим производство кормов.

Введем следующие обозначения: x_j — количество гектаров пашни, отведенной под j -ю кормовую культуру; \underline{d}_j — минимальный размер посевной площади под j -й культурой, способный обеспечить необходимое разнообразие рациона кормления скота; \bar{d}_j — максимальный размер посевной площади, который в силу естественных (почвенно-климатических) условий может быть отведен под j -ю культуру; a_{ij} — выход питательных веществ i -го вида с 1 га посева j -й кормовой культуры; a_i — потребное количество питательных веществ i -го вида, необходимое для выполнения плана по

всей продукции животноводства; b_{kj} — нормы расхода производственных факторов k -го вида для обработки 1 га пашни, отведенной под j -ю кормовую культуру; b_k — ресурсы производственных факторов k -го вида; c_j — себестоимость кормов j -го вида с 1 га посева; σ — общая площадь, отведенная под кормовые культуры. Задача определения оптимальной структуры посевов кормовых культур сводится, таким образом, к определению системы чисел x_j ($j = 1, 2, \dots, n$), обращающих в минимум линейную форму

$$L = \sum_{j=1}^n c_j x_j \quad (6.19)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \geq a_i, \quad i = 1, 2, \dots, m, \quad (6.20)$$

$$\sum_{j=1}^n b_{kj} x_j \leq b_k, \quad k = 1, 2, \dots, s, \quad (6.21)$$

$$\sum_{j=1}^n x_j \leq \sigma, \quad (6.22)$$

$$d_j \leq x_j \leq \bar{d}_j, \quad j = 1, 2, \dots, n. \quad (6.23)$$

Показатель качества (6.19) решения — суммарные затраты на производство кормов. Условия (6.20) обеспечивают требуемое количество каждого из питательных веществ в рационе. Потребности в питательных веществах определяются запланированными поставками по разным видам продукции животноводства. Условия (6.21) фиксируют ограничения по ресурсам различных производственных факторов. Условие (6.22) определяется общей площадью, которая может быть отведена хозяйством под кормовые культуры. Смысл двусторонних ограничений (6.23) не требует пояснений.

6.5. Структура отраслей животноводства и развитие кормовой базы. Определение рациональной структуры отраслей животноводства и планирование развития кормовой базы — связанные задачи. Естественно поэтому совместить решение этих задач.

Сохраним за обозначениями x_j , a_{ij} , b_{kj} , b_k , \underline{d}_j , \bar{d}_j и σ тот же смысл, который придавался этим параметрам в предыдущем пункте. Примем, кроме того, следующие дополнительные обозначения: y_r — количество голов скота r -го вида ($r = 1, 2, \dots, t$); f_{ir} — норма расхода i -го питательного вещества на одну голову скота r -го вида; g_{kr} — норма затрат k -го производственного фактора на 1 голову скота r -го вида; p_r — доход от продукции, отнесенной к 1 голове скота r -го вида; q_j — затраты на получение кормов с 1 га пашни, отведенной под j -ю кормовую культуру.

Ограничения по посевной площади имеют тот же вид, что и в предыдущей задаче:

$$\sum_{j=1}^n x_j \leq \sigma, \quad (6.24)$$

$$\underline{d}_j \leq x_j \leq \bar{d}_j, \quad j = 1, 2, \dots, n. \quad (6.25)$$

Условия, определяемые ограниченными ресурсами различных производственных факторов, записываются в виде

$$\sum_{j=1}^n b_{kj} x_j + \sum_{r=1}^t g_{kr} y_r \leq b_k, \quad k = 1, 2, \dots, s. \quad (6.26)$$

Первое слагаемое в (6.26) соответствует затратам на производство кормов, второе слагаемое определяет непосредственные затраты на животноводство.

В некоторых производственных факторах, используемых при производстве кормов, может не быть необходимости в животноводческой ферме и наоборот. Соответствующие коэффициенты b_{kj} и g_{kr} следует полагать равными нулю.

Условия, отражающие балансовую взаимосвязь между производством кормов и потребностями в них, могут быть представлены в виде

$$\sum_{j=1}^n a_{ij} x_j = \sum_{r=1}^t f_{ir} y_r, \quad i = 1, 2, \dots, m. \quad (6.27)$$

Левая часть i -го равенства представляет собой производство i -го питательного вещества при принятой структуре кормовой базы. Правая часть i -го равенства фиксирует суммарную потребность в i -м питательном веществе для всех видов скота, разводимых в хозяйстве.

Помимо химических удобрений, производство кормов требует органических удобрений. Отсюда дополнительная связь между числом животных и структурой посевов.

Пусть h_j — количество навоза, требующееся на 1 га посевов j -й культуры, а e_r — количество навоза от одной головы скота за плановый период. В этих обозначениях требование по органическим удобрениям записывается так:

$$\sum_{j=1}^n h_j x_j \leq \sum_{r=1}^t e_r y_r. \quad (6.28)$$

Доход хозяйства от развития животноводства равен доходу от продуктов животноводства за вычетом затрат на производство кормов. В принятых обозначениях имеем

$$L = \sum_{r=1}^t p_r y_r - \sum_{j=1}^n q_j x_j. \quad (6.29)$$

Планирование структуры животноводства и посевов кормовых культур сводится, таким образом, к решению следующей задачи линейного программирования. Требуется вычислить значения переменных

$$x_j \quad (j = 1, 2, \dots, n)$$

и

$$y_r \geq 0 \quad (r = 1, 2, \dots, t),$$

обращающих в максимум линейную [форму (6.29) при условиях (6.24) — (6.28)]. Анализ некоторых моделей подобного рода производится в [43].

§ 7. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К ПЛАНИРОВАНИЮ ПРОИЗВОДСТВА

В работе Л. В. Канторовича [18] приводится обобщение производственной задачи, которая рассматривалась в § 5 гл. 1, посвященном экономической интерпретации задачи линейного программирования.

Производство или группа предприятий должны производить r различных продуктов в заранее заданной пропорции (или, как говорят обычно, с учетом заданного ассортимента). Выпуск продукции требует затрат рабочей силы различной квалификации, сырья, энергии и других производственных факторов. Производственные факторы могут быть использованы не только для производства конечных продуктов, но и для производства вспомогательного оборудования, инструмента и других промежуточных продуктов. При планировании необходимо учитывать наличные ресурсы производственных факторов, ограничивающие их затраты. Предполагается, что к началу работы оборудование или полуфабрикаты, именуемые здесь промежуточными продуктами, отсутствовали.

При составлении плана исходят из того, что имеется некоторое количество (N) заранее отработанных технологических способов производства, каждый из которых характеризуется тремя векторами — наборами чисел, определяющими объемы производства каждого конечного продукта, отдельных промежуточных продуктов и затраты по каждому производственному фактору.

Исходные способы производства учитывают главным образом технологические возможности и особенности производства и вовсе не связаны с ограничениями и требованиями в каждом отдельном случае. Комбинируя различные способы производства, можно составлять планы, удовлетворяющие конкретным условиям производства, учитывающие наличные ресурсы и обеспечивающие требуемый ассортимент. Из всех этих планов необходимо выбрать лучший — план, гарантирующий наиболее высокий уровень производства.

План, таким образом, представляет собой набор чисел, показывающих, с каким удельным весом следует использовать тот или иной из исходных способов производства.

Переведем задачу о планировании производства на формальный язык.

Пусть для производства r конечных продуктов требуется n видов производственных факторов. При этом могут быть получены m промежуточных продуктов.

Обозначим векторы, характеризующие соответственно производство окончательных продуктов, промежуточных

продуктов и затраты производственных факторов при s -м исходном способе производства, через

$$\begin{aligned} A^{(s)} &= (a_{1s}, a_{2s}, \dots, a_{rs}), \\ B^{(s)} &= (b_{1s}, b_{2s}, \dots, b_{ms}), \\ C^{(s)} &= (c_{1s}, c_{2s}, \dots, c_{ns}), \\ s &= 1, 2, \dots, N. \end{aligned}$$

Компоненты векторов показывают объемы производства конечных продуктов, промежуточных продуктов и производственных факторов. Отрицательные компоненты соответствуют затратам.

Каждый план определяется вектором

$$X = (x_1, x_2, \dots, x_N).$$

Плану X соответствует производство i -го окончательного продукта, равное

$$\sum_{s=1}^N a_{is} x_s, \quad i = 1, 2, \dots, r.$$

Производство j -го промежуточного продукта при плане X равно

$$\sum_{s=1}^N b_{js} x_s, \quad j = 1, 2, \dots, m.$$

Затраты k -го производственного фактора при выбранном плане определяются числами

$$\sum_{s=1}^N c_{ks} x_s, \quad k = 1, 2, \dots, n.$$

Пусть наличие производственных факторов задано вектором

$$-C^{(0)} = -(c_{10}, c_{20}, \dots, c_{n0}).$$

Предполагается, что $c_{k0} > 0$. Тогда $-c_{k0}$ определяет допустимые затраты k -го производственного фактора. Запишем теперь перечисленные выше ограничения, которым удовлетворяет допустимый план,

Условия, ограничивающие затраты производственных факторов наличными запасами, записываются в виде

$$\sum_{s=1}^N c_{ks} x_s \geq -c_{k0}, \quad k=1, 2, \dots, n. \quad (7.1)$$

Условия, исключающие затраты промежуточных продуктов, имеют вид

$$\sum_{s=1}^N b_{js} x_s \geq 0, \quad j=1, 2, \dots, m. \quad (7.2)$$

Остается записать требование ассортиментности продукции.

Пусть комплект продукции состоит из a_{i0} единиц i -го конечного продукта ($i=1, 2, \dots, r$). Показатель качества планирования — получение максимального количества k комплектов. При этом, естественно, должны соблюдаться условия (7.1), (7.2) и система ограничений

$$\frac{\sum_{s=1}^N a_{is} x_s}{a_{i0}} \geq k, \quad i=1, 2, \dots, r.$$

Обозначая $\frac{a_{is}}{a_{i0}}$ через α_{is} и полагая $k = x_{N+1}$, получим систему соотношений

$$\sum_{s=1}^N \alpha_{is} x_s \geq x_{N+1}, \quad i=1, 2, \dots, r. \quad (7.3)$$

Из физического смысла переменных следует, что x_s должны быть неотрицательны, т. е.

$$x_s \geq 0, \quad s=1, 2, \dots, N. \quad (7.4)$$

Задача планирования сводится, таким образом, к определению набора чисел $X=(x_1, x_2, \dots, x_N, x_{N+1})$, при которых соблюдаются условия (7.1)—(7.4), а величина x_{N+1} достигает наибольшего возможного значения. Мы пришли к задаче линейного программирования в форме Л. В. Канторовича.

Как отмечалось в § 3 гл. 1, эта задача может формулироваться также в следующем эквивалентном виде.

Требуется выбрать систему неотрицательных чисел x_j ($j=1, 2, \dots, N$), доставляющих максимум функции

$$V(x_1, x_2, \dots, x_N) = \min \sum_{1 \leq i \leq r} \sum_{s=1}^N \alpha_{is} x_s$$

при условиях (7.1)—(7.2).

§ 8. ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К ПЛАНИРОВАНИЮ ЭКОНОМИКИ

Линейное программирование является одним из наиболее эффективных методов анализа экономических моделей.

8.1. Рассмотрим модель межотраслевых связей экономического района. Пусть экономика района определяется n отраслями производства. План производства конечной продукции каждой отрасли предполагается заданным. Часть конечной продукции вывозится в другие районы, часть потребляется на месте. Все отрасли производства взаимосвязаны, функционирование любой из них возможно лишь, если обеспечены необходимые поставки средств производства из других отраслей. План производства в отдельных отраслях экономики и межотраслевые связи должны обеспечить минимум затрат труда (живого и овеществленного) на производство конечных продуктов в заданном количестве к установленному сроку.

Переведем на формальный язык задачу планирования межотраслевых связей экономического района указанного типа. Введем следующие обозначения: x_i — продукция i -й отрасли хозяйства за рассматриваемый промежуток времени (для определенности за год); x_{ij} — поставка товаров i -й отрасли j -й отрасли за год; y_i — объем выпуска конечной продукции, запланированный i -й отрасли на год; x_{ij} и y_i , так же как и x_i , измеряются в натуральных единицах. Величины y_i определяются государственными народнохозяйственными планами, учитывающими, в частности, потребности района.

Задача планирования экономики района в условиях статической модели заключается в составлении заданий

предприятиям различных отраслей, при которых будут обеспечены требуемые поставки из одной отрасли в другую и заданный объем конечной продукции каждой отрасли.

Пользуясь введенными обозначениями, запишем условия статической задачи в виде системы неравенств

$$x_i \geq \sum_{j=1}^n a_{ij} x_j + y_i. \quad (8.1)$$

Отношение $\frac{x_{ij}}{x_j} = a_{ij}$ определяет количество продукции i -й отрасли, необходимое для производства единицы продукции j -й отрасли. В новых обозначениях система условий (8.1) запишется в виде

$$x_i - \sum_{j=1}^n a_{ij} x_j \geq y_i. \quad (8.2)$$

Экономика является относительно инерционной системой. Исходя из этого, В. Леонтьев [28] предложил использовать для расчетов на будущий промежуток времени величины a_{ij} , систематизированные за прошедший промежуток.

Соотношения (8.2) связывают плановые объемы производства каждой отрасли с заданными размерами конечной продукции исследуемой экономической системы. Размеры производства x_i в каждой отрасли ограничиваются также наличными на данный период мощностями оборудования, лимитируемыми материалами, полуфабрикатами, энергией, рабочей силой и другими факторами.

Пусть величина d_{ik} определяет предельно допустимый уровень производства в i -й отрасли хозяйства при наличии ограничений только по k -му фактору. Это значит, что объем продукции i -й отрасли производства за рассматриваемый промежуток времени должен также удовлетворять неравенствам

$$0 \leq x_i \leq d_i, \quad i = 1, 2, \dots, n, \quad (8.3)$$

где $d_i = \min_k d_{ik}$. Если производственные мощности v -й отрасли не ограничены, полагаем $d_v = \infty$.

Пусть себестоимость производства одной единицы продукции j -й отрасли измеряется величиной c_j . Тогда общие народнохозяйственные затраты рассматриваемой экономической системы равны

$$L = \sum_{j=1}^n c_j x_j. \quad (8.4)$$

Планирование развития экономического района сводится, таким образом, к решению задачи линейного программирования, в которой требуется минимизировать линейную форму (8.4) при условиях (8.2) и (8.3). Исследование, проведенное в [51], показало, однако, что при естественных с экономической точки зрения допущениях о параметрах a_{ij} задача (8.2)—(8.4) либо вовсе не имеет решения, либо имеет решение, не зависящее от размеров затрат c_j . Примем, что

$$\sum_{i=1}^n a_{ij} < 1, \quad j = 1, 2, \dots, n, \quad (8.5)$$

или, что то же самое,

$$\sum_{i=1}^n x_{ij} < x_j, \quad j = 1, 2, \dots, n. \quad (8.6)$$

Последнее соотношение означает, что затраты труда на продукцию, производимую j -й отраслью, превышают затраты труда на средства производства, поставляемые ей из других отраслей. (Заметим, что в предыдущих соотношениях x_j и x_{ij} могли измеряться в натуральных единицах или в деньгах как выражения затрат различного качества труда. В неравенствах (8.6) можно говорить только о денежном выражении продукции. Иначе нельзя сопоставлять и складывать продукцию различных отраслей.)

Вывод, полученный в работе [51], справедливый при выполнении условий (8.6), может быть сформулирован следующим образом. Составляющие x_j^* решения $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ системы уравнений

$$x_i - \sum_{j=1}^n a_{ij} x_j = y_i, \quad i = 1, 2, \dots, n, \quad (8.7)$$

являются неотрицательными величинами. Если x_j^* удовлетворяют, кроме того, дополнительным ограничениям

$$x_j^* \leq d_j, \quad j = 1, 2, \dots, n,$$

то они оказываются компонентами решения задачи (8.2)—(8.4) при любых величинах затрат c_j . В противном случае задача не имеет решения.

8.2. В практике планирования производственные ограничения вида (8.3), как правило, являются существенными ограничениями. Обычно условия (8.3) противоречат системе (8.7), т. е. не существует плана, который при фиксированных ограничениях (8.3) и заданных коэффициентах a_{ij} межотраслевых связей обеспечил бы выпуск требуемой конечной продукции. В таких случаях постановка задачи планирования экономической системы должна быть изменена.

Если имеются дополнительные ресурсы, естественно рассмотреть вопрос о том, как их наиболее экономным путем использовать, чтобы, расширив производственные возможности и ослабив таким образом ограничения (8.3), обеспечить выпуск требуемой продукции. Анализ таких моделей требует методов кусочно-линейного программирования (см. также [12]). Ниже описаны модели, которые исключают возможность использования дополнительных ресурсов, позволяющих ослабить ограничения (8.3). В этом случае задача экономного планирования сводится к определению таких объемов производства x_j каждой отрасли хозяйства, при которых ущерб от недополучения требуемого количества конечной продукции был бы минимальным.

Возможны следующие подходы к определению ущерба от недополучения требуемого объема конечной продукции.

1. Ущерб Q характеризуется средневзвешенной величиной невыполненных поставок конечных продуктов по разным отраслям:

$$Q = \sum_{i=1}^n g_i \left\{ y_i - \left(x_i - \sum_{j=1}^n a_{ij} x_j \right) \right\},$$

где $g_i \geq 0$ ($i = 1, 2, \dots, n$) — весовые коэффициенты.

Выбор коэффициентов g_i — самостоятельная задача, лежащая вне пределов экономики, но опирающаяся на экономические характеристики. Решение этой задачи определяется ролью, которая отводится в более широком плане поставкам конечных продуктов каждой отрасли района; g_i — размерные коэффициенты, позволяющие выразить ущерб от недополучения конечной продукции разных отраслей в эквивалентных единицах.

В данном случае задача об определении наиболее экономного плана производства формулируется следующим образом.

Требуется определить переменные x_j , которые обращают в минимум линейную форму

$$Q = \sum_{i=1}^n g_i \left(y_i - x_i + \sum_{j=1}^n a_{ij} x_j \right) \quad (8.8)$$

при условиях

$$x_i - \sum_{j=1}^n a_{ij} x_j \leq y_i, \quad i = 1, 2, \dots, n, \quad (8.9)$$

$$0 \leq x_j \leq d_j, \quad j = 1, 2, \dots, n. \quad (8.10)$$

Ограничения (8.9) выражают то естественное условие, что при ограниченных ресурсах, исключающих возможность выполнения поставок по всем отраслям, нецелесообразно стремиться к превышению требуемого объема продукции по отдельным отраслям. $\sum_{i=1}^n g_i y_i$ — фиксированная величина. Поэтому задача (8.8) — (8.10) сводится к вычислению набора переменных x_j ($j = 1, 2, \dots, n$), который обращает в максимум линейную форму

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

при условиях

$$\begin{aligned} x_i - \sum_{j=1}^n a_{ij} x_j &\leq y_i, & i = 1, 2, \dots, n, \\ 0 &\leq x_j \leq d_j, & j = 1, 2, \dots, n. \end{aligned}$$

Здесь

$$c_j = g_j - \sum_{i=1}^n g_i a_{ij}.$$

2. В некоторых ситуациях может оказаться, что существу задачи будет более соответствовать другое определение ущерба. Именно под ущербом подразумевается величина

$$L = \max_i g_i \left[y_i - \left(x_i - \sum_{j=1}^n a_{ij} x_j \right) \right].$$

Другими словами, ущерб определяется отраслью, в которой разность между требуемым и обеспечиваемым объемом конечной продукции в эквивалентных единицах, основанных на указанных выше весовых коэффициентах, максимальна.

Экономный план обращает в минимум ущерб, наносимый хозяйству ограничениями производственных возможностей отдельных отраслей экономики. Следовательно, экономный план определяется набором чисел x_j ($j = 1, 2, \dots, n$), на котором достигается

$$\min L = \min_x \max_i g_i \left[y_i - \left(x_i - \sum_{j=1}^n a_{ij} x_j \right) \right]$$

при условиях

$$x_i - \sum_{j=1}^n a_{ij} x_j \leq y_i, \quad i = 1, 2, \dots, n,$$

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

Нетрудно видеть (см., например, предыдущий параграф), что это обычная задача линейного программирования. Действительно, последняя задача эквивалентна следующей.

Требуется обратить в минимум

$$L = x_{n+1} \quad (8.11)$$

при условиях

$$0 \leq y_i - x_i + \sum_{j=1}^n a_{ij}x_j \leq \frac{1}{g_i} x_{n+1}, \quad i = 1, 2, \dots, n, \quad (8.12)$$

$$0 \leq x_j \leq d_j, \quad j = 1, 2, \dots, n. \quad (8.13)$$

Условия (8.12) означают, что ни в одной отрасли производство не превышает требуемого объема продукции, а размеры невыполненных поставок конечных продуктов в эквивалентных единицах ни в одной отрасли не превосходят некоторой величины x_{n+1} , определяющей ущерб народного хозяйства из-за ограниченных ресурсов.

Заметим, что ограничения по производственным возможностям различных отраслей могут задаваться не только непосредственно в форме (8.13), но и предельными уровнями факторов, необходимых ряду отраслей или всем отраслям производства. Так, например, можно учитывать ограничения по электроэнергии, топливу, транспорту, оборудованию, кадрам и т. д. Подобные ограничения могут записываться в виде

$$\sum_{j=1}^n u_{sj}x_j \leq v_s, \quad s = 1, 2, \dots, r, \quad (8.14)$$

где u_{sj} — количество s -го фактора (энергии, горючего и т. д.), необходимое для производства единицы продукции j -й отрасли экономики; v_s — ресурсы s -го фактора, которыми располагает район.

Таким образом, планирование экономической системы, в которой исключена возможность превышения фиксированных ограничений по ресурсам в течение планируемого периода, сводится к следующей задаче линейного программирования.

Требуется вычислить переменные $x_1, x_2, \dots, x_n, x_{n+1}$, доставляющие минимум величине

$$L = x_{n+1}$$

при условиях

$$0 \leq y_i - x_i + \sum_{j=1}^n a_{ij}x_j \leq \frac{1}{g_i} x_{n+1}, \quad i = 1, 2, \dots, n,$$

$$\sum_{j=1}^n u_{sj}x_j \leq v_s, \quad s = 1, 2, \dots, r,$$

$$0 \leq x_j \leq d_j, \quad j = 1, 2, \dots, n,$$

§ 9. ОБ ОПТИМАЛЬНОМ РАСКРОЕ МАТЕРИАЛОВ

Одна из первых прикладных задач, потребовавших упорядочения перебора вариантов, — это так называемая «задача фанерного треста». По-видимому, именно эта задача положила начало развитию методов линейного программирования в Советском Союзе. Задача фанерного треста в дальнейшем не раз обобщалась и может быть сформулирована как задача о максимальном сокращении отходов или задача об оптимальном раскрое материалов.

Полуфабрикаты поступают на предприятие в виде листов фанеры, жести, стали, стекла. Из полуфабрикатов требуется изготовить возможно большее число комплектов деталей. При этом должны быть соблюдены следующие условия. Всего имеется n партий материала, причем i -я партия содержит q_i единиц. Комплект состоит из m разных деталей. В комплект входит p_k деталей k -го типа. Единица каждой партии может раскраиваться s различными способами.

Пусть при j -м способе раскроя одной единицы i -й партии полуфабрикатов получается a_{ikj} деталей k -го типа.

Обозначим через x_{ij} количество единиц i -й партии материалов, которые следует раскроить по j -му способу. Количество деталей k -го типа, которое будет при этом получено, равно $a_{ikj}x_{ij}$. Количество деталей k -го типа, которое можно получить из i -й партии полуфабрикатов, используя все способы раскроя, равно $\sum_{j=1}^s a_{ikj}x_{ij}$. Общее число деталей k -го типа, очевидно, можно получить, если сложить количества деталей этого вида, выкраиваемых из каждой партии материалов:

$$\sum_{j=1}^s a_{1kj}x_{1j} + \sum_{j=1}^s a_{2kj}x_{2j} + \dots + \sum_{j=1}^s a_{nkj}x_{nj} = \sum_{i=1}^n \sum_{j=1}^s a_{ikj}x_{ij}.$$

Каждый комплект содержит p_k деталей k -го типа. Поэтому количество комплектов, обеспеченных деталями k -го типа, равно

$$z_k = \frac{\sum_{i=1}^n \sum_{j=1}^s a_{ikj}x_{ij}}{p_k}.$$

Комплект должен быть обеспечен деталями всех типов. Это значит, что задача об оптимальном раскрое материалов сводится к вычислению набора чисел x_{ij} ($i=1, 2, \dots, n, j=1, 2, \dots, s$), при котором обеспечивается максимум минимального из отношений z_k ($k=1, 2, \dots, m$). Другими словами, необходимо обеспечить максимум z при условиях

$$z_k \geq z, \quad k=1, 2, \dots, m. \quad (9.1)$$

Кроме того, должны быть учтены условия

$$\sum_{j=1}^s x_{ij} = q_i, \quad i=1, 2, \dots, n, \quad (9.2)$$

$$x_{ij} \geq 0, \quad i=1, 2, \dots, n, \quad j=1, 2, \dots, s. \quad (9.3)$$

В первой группе условий записано, что i -я партия содержит q_i единиц материала. Вторая группа неравенств не требует пояснения.

ЭЛЕМЕНТЫ ТЕОРИИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В этой главе дается краткое изложение теории линейного программирования. Первый параграф носит вспомогательный характер и содержит в основном некоторые сведения из теории выпуклых множеств, необходимые для дальнейшего. Следующий параграф посвящен теории выпуклых многогранных множеств, на которой базируется обоснование ряда важных теорем линейного программирования (§ 3) и геометрия линейного программирования (§ 4). В последних трех параграфах главы изучаются двойственные задачи линейного программирования, связывающие их соотношения двойственности и вытекающие из этих соотношений критерии оптимальности.

§ 1. НЕОБХОДИМЫЕ СВЕДЕНИЯ ИЗ ЛИНЕЙНОЙ АЛГЕБРЫ И ТЕОРИИ ВЫПУКЛЫХ МНОЖЕСТВ

1.1. Упорядоченная система n действительных чисел

$$X = (x_1, x_2, \dots, x_n)$$

называется n -мерным вектором. Числа x_1, x_2, \dots, x_n будем называть *компонентами* или *составляющими* вектора X . Векторы будем обычно обозначать большими буквами, а их компоненты — соответствующими малыми буквами с индексами, указывающими на порядковый номер компоненты.

Два n -мерных вектора

$$X = (x_1, x_2, \dots, x_n) \quad \text{и} \quad Y = (y_1, y_2, \dots, y_n)$$

считаются равными, если их соответствующие компоненты совпадают, т. е. если $x_i = y_i$ для $i = 1, 2, \dots, n$.

Суммой $X + Y$ векторов X и Y называется вектор

$$Z = (z_1, z_2, \dots, z_n)$$

с компонентами

$$z_i = x_i + y_i, \quad i = 1, 2, \dots, n.$$

Как нетрудно проверить, сложение векторов обладает свойствами коммутативности

$$X + Y = Y + X$$

и ассоциативности

$$(X + Y) + Z = X + (Y + Z).$$

Разностью упорядоченной пары векторов X и Y называется вектор Z вида

$$Z = X - Y = (x_1 - y_1, x_2 - y_2, \dots, x_n - y_n).$$

Вектор, все компоненты которого равны нулю, называется *нулевым вектором*:

$$0 = (0, 0, \dots, 0).$$

Таким образом, для равенства векторов X и Y необходимо и достаточно, чтобы

$$X - Y = 0.$$

Произведением вектора X на действительное число (скаляр) α называется вектор

$$\alpha X = (\alpha x_1, \alpha x_2, \dots, \alpha x_n).$$

Из приведенных определений суммы двух векторов и произведения вектора на скаляр непосредственно вытекают следующие свойства этих операций:

$$\alpha_1 (\alpha_2 X) = (\alpha_1 \alpha_2) X \quad (\text{ассоциативность}),$$

$$\left. \begin{aligned} (\alpha_1 + \alpha_2) X &= \alpha_1 X + \alpha_2 X, \\ \alpha (X + Y) &= \alpha X + \alpha Y \end{aligned} \right\} \quad (\text{дистрибутивность}),$$

$$0 \cdot X = \alpha \cdot 0 = 0.$$

Совокупность всех n -мерных векторов, для которых введены понятия сложения, вычитания и умножения на скаляр, называется *n -мерным векторным пространством (действительным)*. Слово «действительное», стоящее в скобках, указывает на то, что компоненты векторов и скаляры являются действительными числами. В дальней-

шем будем его опускать. Однако это не должно привести к недоразумению, так как мы будем иметь дело только с действительным случаем.

При $n=3$ введенное векторное пространство соответствует обычному пространству трех измерений. Каждой упорядоченной тройке чисел (x_1, x_2, x_3) в этом пространстве соответствует точка с координатами x_1, x_2, x_3 (или вектор, направленный в нее из начала координат). Наоборот, каждой точке трехмерного пространства может быть сопоставлена упорядоченная тройка действительных чисел, составленная из координат этой точки. Таким образом, элементы n -мерного векторного пространства при $n=3$ можно интерпретировать либо как точки, либо как векторы, проведенные из начала координат. То же самое относится к случаю $n=2$ (плоский случай). В дальнейшем слова «вектор» и «точка» будут использоваться как синонимы для обозначения элементов векторного пространства.

1.2. Каждой паре векторов

$$A = (a_1, a_2, \dots, a_n) \text{ и } B = (b_1, b_2, \dots, b_n)$$

может быть сопоставлено число

$$(A, B) = \sum_{i=1}^n a_i b_i,$$

называемое их *скалярным произведением*.

Введенная функция пары векторов обладает следующими свойствами:

- 1) $(A, B) = (B, A)$;
- 2) $(A_1 + A_2, B) = (A_1, B) + (A_2, B)$;
- 3) $(\lambda A, B) = \lambda (A, B)$;
- 4) $(A, A) \geq 0$, причем знак равенства в этом соотношении возможен лишь при $A=0$;
- 5) $(A, B)^2 \leq (A, A)(B, B)$.

Последнее неравенство обычно связывают с именем Буняковского.

Два вектора A и B называются *ортogonalными*, если $(A, B) = 0$. В частности, нулевой вектор ортogonalен любому вектору данного пространства. Используя понятие скалярного произведения, можно определить

расстояние между точками n -мерного пространства (задать в пространстве метрику).

Длиной или нормой вектора

$$A = (a_1, a_2, \dots, a_n)$$

называется число

$$|A| = +\sqrt{(A, A)} = +\sqrt{\sum_{i=1}^n a_i^2}.$$

В соответствии со свойством 4) скалярного произведения $|A| \geq 0$ для любого вектора A , причем равенство $|A| = 0$ возможно только при $A = 0$.

Расстояние между точками A и B полагается равным

$$\rho(A, B) = |A - B| = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}.$$

Пусть A , B и C — произвольные точки n -мерного пространства. Используя свойства скалярного произведения, можно доказать справедливость следующего неравенства (неравенства треугольника):

$$|A - B| \leq |A - C| + |C - B|.$$

Неравенство Буняковского и неравенство треугольника имеют прозрачный геометрический смысл при n , равном 2 и 3. Известно, что скалярное произведение двух векторов на плоскости или в пространстве (трехмерном) равно произведению их длин, умноженному на косинус угла между ними. Следовательно, неравенство Буняковского, будучи переписано в эквивалентной форме

$$\frac{(A, B)}{|A| \cdot |B|} \leq 1,$$

означает в данном случае, что косинус любого угла не превосходит единицы. Что касается неравенства треугольника, то оно расшифровывается так: сумма длин двух сторон треугольника не меньше длины третьей стороны.

n -мерное векторное пространство, в котором определено понятие скалярного произведения (а следовательно, введена метрика), принято называть *евклидовым*.

В дальнейшем n -мерное евклидово пространство будет обозначаться через E_n .

Говорят, что последовательность векторов $X_1, X_2, \dots, X_k, \dots$ из E_n *сходится* к вектору X , принадлежащему E_n ($\lim_{k \rightarrow \infty} X_k = X$), если для любого $\varepsilon > 0$ существует натуральное число $N(\varepsilon)$, начиная с которого ($i \geq N(\varepsilon)$)

$$|X - X_i| < \varepsilon.$$

Из определения нормы следует, что последовательность $\{X_k\}$ сходится к X в том и только в том случае, когда

$$\lim_{k \rightarrow \infty} x_s^{(k)} = x_s, \quad s = 1, 2, \dots, n.$$

Здесь

$$X_k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \quad k = 1, 2, \dots, \\ X = (x_1, x_2, \dots, x_n).$$

1.3. Пусть A_1, A_2, \dots, A_s — произвольная система n -мерных векторов.

Вектор

$$\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_s A_s,$$

где α_i — скаляры ($i = 1, 2, \dots, s$) называют обычно *линейной комбинацией* векторов A_1, A_2, \dots, A_s с коэффициентами $\alpha_1, \alpha_2, \dots, \alpha_s$.

Система векторов A_1, A_2, \dots, A_s называется *линейно зависимой*, если нулевой вектор

$$0 = \underbrace{(0, 0, \dots, 0)}_n$$

может быть представлен в виде линейной комбинации векторов A_1, A_2, \dots, A_s , среди коэффициентов которой имеются ненулевые.

В противном случае система векторов A_1, A_2, \dots, A_s по определению линейно независима. Другими словами, систему A_1, A_2, \dots, A_s называют *линейно независимой*, если соотношение

$$\sum_{i=1}^s \alpha_i A_i = 0$$

возможно лишь при $\alpha_1 = \alpha_2 = \dots = \alpha_s = 0$. Векторы, составляющие линейно независимую систему, называются *линейно независимыми*.

Нетрудно проверить, что если данный вектор представим в виде линейной комбинации линейно независимых векторов, то это представление единственно.

Приведем критерий линейной независимости системы векторов, широко используемый в различных разделах математики.

Для линейной независимости системы векторов $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$, $i = 1, 2, \dots, s$, необходимо и достаточно существование квадратной матрицы порядка s

$$\begin{vmatrix} a_{1j_1} & a_{1j_2} & \dots & a_{1j_s} \\ a_{2j_1} & a_{2j_2} & \dots & a_{2j_s} \\ \dots & \dots & \dots & \dots \\ a_{sj_1} & a_{sj_2} & \dots & a_{sj_s} \end{vmatrix},$$

составленной из координат векторов A_i , определитель которой отличен от нуля.

Из приведенного критерия следует, что в n -мерном пространстве не существует линейно независимой системы, состоящей более чем из n векторов. С другой стороны, единичные векторы $e_1 = (1, 0, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots, 0)$, \dots , $e_n = (0, 0, 0, \dots, 1)$ составляют, очевидно, линейно независимую систему.

По аналогии с плоскостью и трехмерным пространством *размерностью* векторного пространства естественно называть величину, совпадающую с максимальным числом линейно независимых векторов этого пространства. Таким образом, размерность n -мерного векторного пространства равна n . Это обстоятельство послужило поводом для названия данного пространства *n -мерным*.

Рассмотрим произвольную прямоугольную матрицу A . Матрица, составленная из элементов, расположенных на пересечении фиксированной системы строк и столбцов A , называется *подматрицей* матрицы A . Определитель произвольной квадратной подматрицы матрицы A принято называть *минором* этой матрицы. Порядок минора определяется порядком соответствующей подматрицы.

Наивысший порядок отличных от нуля миноров матрицы называется *рангом* этой матрицы.

Из критерия линейной независимости системы векторов следует, что максимальное число линейно независимых строк матрицы A равно максимальному числу линейно независимых столбцов этой матрицы и совпадает с рангом A .

Рангом произвольной системы векторов называется максимальное число линейно независимых векторов данной системы. Совокупность линейно независимых векторов системы, число которых равно ее рангу, принято называть *базисом* системы. Произвольная совокупность линейно независимых векторов данной системы может быть дополнена до ее базиса. Множество линейно независимых векторов данной системы является ее базисом в том и только в том случае, если любой вектор системы представим в виде линейной комбинации векторов рассматриваемого множества.

1.4. Непустое множество n -мерных векторов назовем *подпространством* n -мерного векторного пространства, если результаты сложения любых двух векторов множества и умножения произвольного вектора множества на скаляр принадлежат этому множеству.

Очевидно, что любое подпространство содержит нулевой вектор, так как $0 \cdot X = 0$ для произвольного элемента X из подпространства. Подпространство называется *r -мерным*, если максимальное число линейно независимых векторов, содержащихся в нем, равно r . Пусть A_1, A_2, \dots, A_r — система линейно независимых векторов n -мерного векторного пространства. Рассмотрим множество всевозможных линейных комбинаций этой системы. Легко показать, что оно является r -мерным подпространством. С другой стороны, в любом r -мерном подпространстве существуют r линейно независимых векторов, совокупность линейных комбинаций которых совпадает с данным подпространством. Итак, r -мерное подпространство можно определить как совокупность всех линейных комбинаций некоторой линейно независимой системы из r векторов. Если $r = 0$, то подпространство состоит из одного лишь нулевого вектора.

Если $r=n$, то оно совпадает со всем пространством. При $0 < r < n$ получаем промежуточные подпространства, содержащие бесчисленные множества элементов и несовпадающие со всем пространством. Если обратиться к обычному пространству трех измерений, то одномерными его подпространствами являются прямые, проходящие через начало координат, а двумерными — плоскости, содержащие начало координат.

Множество элементов n -мерного пространства, представимых в виде

$$X + X',$$

где X — фиксированный вектор, а X' принадлежит некоторому r -мерному подпространству, называется *r -мерным линейным многообразием* ($r \leq n$).

Таким образом, r -мерное линейное многообразие может рассматриваться как некоторый сдвиг r -мерного подпространства. В частности, если сдвиг (вектор X) равен нулевому вектору, то многообразие превращается в подпространство.

В трехмерном пространстве одномерное и двумерное линейные многообразия являются прямой и плоскостью соответственно. Обобщая понятия прямой и плоскости на случай n -мерного пространства, назовем одномерное линейное многообразие *прямой*, $(n-1)$ -мерное линейное многообразие — *гиперплоскостью*.

Учитывая определение r -мерного подпространства, можно сказать, что r -мерное линейное многообразие представляет собой множество точек вида

$$X + \sum_{i=1}^r \alpha_i X_i,$$

где X — фиксированная точка n -мерного пространства; X_1, X_2, \dots, X_r — некоторая линейно независимая система n -мерных векторов, а $\alpha_1, \alpha_2, \dots, \alpha_r$ — произвольные числа.

Иногда удобно бывает использовать иное определение линейного многообразия.

Рассмотрим одно линейное уравнение с n неизвестными

$$\sum_{j=1}^n a_j x_j = b. \quad (1.1)$$

Гиперплоскостью называется множество точек n -мерного пространства, удовлетворяющих линейному уравнению вида (1.1), в котором $A = (a_1, a_2, \dots, a_n) \neq 0$.

Уравнение (1.1) принято называть *уравнением гиперплоскости*. Вектор $A \neq 0$ называется *направляющим вектором* гиперплоскости, определяемой уравнением (1.1).

Пусть X_0 — произвольная точка гиперплоскости (1.1) (гиперплоскости, определяемой уравнением (1.1)). Очевидно, гиперплоскость (1.1) может рассматриваться как совокупность точек, представимых в виде $X_0 + X'$, где $(A, X') = 0$.

Таким образом, гиперплоскость (1.1) является сдвигом на вектор X_0 множества векторов X' , ортогональных направляющему вектору A . Поэтому принято говорить, что гиперплоскость ортогональна своему направляющему вектору.

— Будем говорить, что гиперплоскости *линейно независимы*, если их направляющие векторы образуют линейно независимую систему.

Из теории систем линейных уравнений следует эквивалентность обоих определений гиперплоскости. Используя эту теорию, можно получить также эквивалентное определение произвольного линейного многообразия:

r -мерным линейным многообразием называется общая часть (пересечение) $n - r$ линейно независимых гиперплоскостей (в том случае, если она не пуста).

Итак, r -мерное линейное многообразие можно рассматривать либо как сдвинутое множество всевозможных линейных комбинаций r линейно независимых векторов, либо как общую часть $n - r$ линейно независимых гиперплоскостей.

1.5. В п. 1.2 было введено n -мерное евклидово пространство E_n . Мы будем рассматривать различные множества точек (векторов) пространства E_n . Напомним, что условие принадлежности всех точек множества G_1 множеству G_2 записывается обычно так: $G_1 \subseteq G_2$ (G_1 содержится в G_2). Если множество G_1 содержит только одну точку X , то условие принадлежности этой точки множеству G_2 обозначается по-другому: $X \in G_2$.

В частности, запись $G \subseteq E_n$ означает, что множество G состоит из точек пространства E_n .

Рассмотрим произвольную гиперплоскость

$$(\Lambda, X) = \sum_{i=1}^n \lambda_i x_i = c. \quad (1.2)$$

Гиперплоскость (1.2) порождает пару множеств, называемых *полупространствами*:

$$(\Lambda, X) \leq c, \quad (\Lambda, X) \geq c. \quad (1.3)$$

Уравнение произвольной прямой пространства E_n имеет вид

$$X = A + Bt, \quad -\infty < t < \infty, \quad (1.4)$$

где A и B — некоторые векторы из E_n . Вектор B принято называть *направляющим вектором* данной прямой. Если параметр t в уравнении (1.4) ограничен снизу или сверху конечным числом, то соответствующее множество называется *лучом* (или *полупрямой*) с направляющим вектором B .

При наличии двусторонних конечных ограничений на параметр t уравнение (1.4) определяет *отрезок*. Легко убедиться, что отрезок с концами в точках A' и A'' является совокупностью точек X вида

$$X = \mu A' + (1 - \mu) A'', \quad 0 \leq \mu \leq 1.$$

Шаром с центром в точке $A \in E_n$ и радиусом $\rho > 0$ называется множество точек $X \in E_n$, для которых

$$|X - A| < \rho. \quad (1.5)$$

Шар (1.5) составляет ρ -окрестность точки A .

Если множество $G \subseteq E_n$ содержит вместе с точкой X ее ε -окрестность при некотором $\varepsilon > 0$, то X называется *внутренней точкой* множества G . *Внешняя* точка множества G обладает некоторой окрестностью, находящейся вне G . Если в любой окрестности точки X содержатся как точки G , так и точки, не принадлежащие G , то по определению X — *граничная точка* множества G .

Множество, содержащее все свои граничные точки, называются *замкнутым*.

Нетрудно убедиться, что линейное многообразие (и, в частности, гиперплоскость и прямая), полупространство, луч, отрезок являются замкнутыми множествами. Легко проверить также, что объединение конечного числа замкнутых множеств G_1, G_2, \dots, G_k (обозначается $\bigcup_{i=1}^k G_i$) и общая часть произвольного числа замкнутых множеств G_1, G_2, \dots (обозначается $\bigcap_i G_i$) являются замкнутыми множествами.

Множество G называется *ограниченным*, если найдется такое число c , не зависящее от X , что

$$|X| < c$$

для всех $X \in G$.

На n -мерное евклидово пространство без труда переносится известная теорема Больцано—Вейерштрасса:

Из любой ограниченной бесконечной последовательности точек пространства E_n может быть выделена сходящаяся подпоследовательность.

Множество $G \subseteq E_n$ называется *выпуклым*, если вместе с любыми двумя своими точками A и B оно содержит соединяющий их отрезок

$$\mu A + (1 - \mu) B, \quad 0 \leq \mu \leq 1.$$

Все упоминавшиеся в этом пункте множества являются выпуклыми.

Нетрудно проверить, что если G — выпуклое множество и точки P_1, P_2, \dots, P_s содержатся в G , то при любых $\alpha_i \geq 0, i = 1, 2, \dots, s$, удовлетворяющих условию

$$\sum_{i=1}^s \alpha_i = 1,$$

точка $\sum_{i=1}^s \alpha_i P_i \in G$.

Нетрудно проверить также, что общая часть произвольного числа выпуклых множеств является выпуклым множеством.

Важный класс выпуклых множеств составляют так называемые выпуклые конусы. Приведем соответствующее определение.

Будем говорить, что выпуклое замкнутое множество T является *выпуклым конусом* с вершиной в точке P_0 , если для произвольного вектора $P \in T$ и любого $\mu \geq 0$ вектор

$$P_0 + \mu(P - P_0) \in T.$$

Нетрудно проверить, что всякое линейное многообразие является выпуклым конусом, причем за вершину конуса может быть принята любая точка многообразия.

Другим достаточно общим примером выпуклого конуса является следующий. Пусть D — произвольное выпуклое замкнутое ограниченное множество, а P_0 — некоторая точка, расположенная вне D . Совокупность всех лучей, исходящих из P_0 и пересекающих D , является выпуклым конусом.

1.6. Мы сейчас сформулируем и докажем одно фундаментальное утверждение относительно выпуклых множеств. Это утверждение является основой доказательства многих важных фактов теории линейного и нелинейного программирования.

Начнем с определения. Будем говорить, что гиперплоскость, определяемая уравнением

$$(\Lambda, X) = c,$$

разделяет множества G_1 и G_2 , если

$$(\Lambda, X) \leq c \quad \text{для всех точек} \quad X \in G_1,$$

$$(\Lambda, X) \geq c \quad \text{для всех точек} \quad X \in G_2.$$

Другими словами, некоторая гиперплоскость разделяет множества G_1 и G_2 в том и только в том случае, если G_1 расположено в одном из полупространств, порождаемых гиперплоскостью, а G_2 лежит в другом полупространстве.

Если все неравенства, участвующие в определении разделяющей гиперплоскости, являются строгими, то говорят, что данная гиперплоскость *строго разделяет* множества G_1 и G_2 .

Таким образом, гиперплоскость, строго разделяющая множества G_1 и G_2 , обладает следующим свойством. Все

точки множества G_1 являются внутренними точками одного из полупространств, порождаемых гиперплоскостью, а все точки множества G_2 оказываются внутренними точками другого полупространства.

Теорема 1.1 (теорема о разделяющей гиперплоскости). Пусть G_1 и G_2 — произвольные выпуклые замкнутые множества без общих точек, из которых хотя бы одно ограничено. В этих предположениях существует гиперплоскость, строго разделяющая множества G_1 и G_2 .

Доказательство. Для лучшего усвоения приводимых ниже рассуждений полезно сопоставить их с рис. 3.1, где указана геометрия основных моментов доказательства при $n=2$.

1. Для определенности допустим, что ограниченным множеством является G_1 . Положим

$$\sigma = \inf |P_1 - P_2|,$$

где нижняя грань (\inf) берется по всевозможным точкам $P_1 \in G_1$ и $P_2 \in G_2$. Согласно определению нижней грани существуют такие последовательности точек $\{P_1^{(k)}\}$ и $\{P_2^{(k)}\}$ ($P_1^{(k)} \in G_1$, $P_2^{(k)} \in G_2$), что

$$\lim_{k \rightarrow \infty} |P_1^{(k)} - P_2^{(k)}| = \sigma. \quad (1.6)$$

Поскольку G_1 — ограниченное множество, последовательность $\{P_1^{(k)}\}$ ограничена. Последовательность $\{P_2^{(k)}\}$ также ограничена, так как

$$|P_2^{(k)}| = |(P_2^{(k)} - P_1^{(k)}) + P_1^{(k)}| \leq |P_1^{(k)} - P_2^{(k)}| + |P_1^{(k)}|.$$

Пользуясь теоремой Больцано—Вейерштрасса, выделим из последовательностей $\{P_1^{(k)}\}$ и $\{P_2^{(k)}\}$ подпоследовательности, сходящиеся соответственно к точкам P_1^* , P_2^* . В силу замкнутости рассматриваемых множеств $P_1^* \in G_1$, $P_2^* \in G_2$.

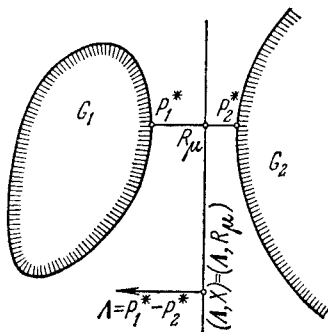


Рис. 3.1.

Таким образом, существует последовательность индексов k_i , $i = 1, 2, \dots$, для которой

$$\lim_{i \rightarrow \infty} |P_1^{(k_i)} - P_1^*| = 0, \quad \lim_{i \rightarrow \infty} |P_2^{(k_i)} - P_2^*| = 0.$$

Переходя в выражении $|P_1^{(k_i)} - P_2^{(k_i)}|$ к пределу при $i \rightarrow \infty$ и учитывая (1.6), получаем

$$|P_1^* - P_2^*| = \sigma = \min_{\substack{P_1 \in G_1 \\ P_2 \in G_2}} |P_1 - P_2|. \quad (1.7)$$

По условию множества G_1 , G_2 не имеют общих точек; следовательно, $\sigma > 0$.

2. Пусть

$$R_\mu = \mu P_1^* + (1 - \mu) P_2^*,$$

где $0 < \mu < 1$. Обозначим через Π_μ гиперплоскость, определяемую уравнением

$$(\Lambda, X) = (\Lambda, R_\mu), \quad (1.8)$$

где $\Lambda = P_1^* - P_2^*$.

Непосредственной подстановкой убеждаемся, что Π_μ содержит точку R_μ . Покажем, что гиперплоскость Π_μ строго разделяет множества G_1 и G_2 . Доказательство проводится от противного.

Пусть $Q \in G_1$, и вместе с тем

$$(\Lambda, Q) \leq (\Lambda, R_\mu). \quad (1.9)$$

Поскольку P_1^* и Q принадлежат выпуклому множеству G_1 , любая точка $Q_\varepsilon = (1 - \varepsilon) P_1^* + \varepsilon Q$, где $0 \leq \varepsilon \leq 1$, также содержится в G_1 .

Вычислим квадрат расстояния между точками $Q_\varepsilon \in G_1$ и $P_2^* \in G_2$. Имеем

$$\begin{aligned} |Q_\varepsilon - P_2^*|^2 &= |(1 - \varepsilon) P_1^* + \varepsilon Q - P_2^*|^2 = |\Lambda + \varepsilon (Q - P_1^*)|^2 = \\ &= \varepsilon^2 |Q - P_1^*|^2 + 2\varepsilon (\Lambda, Q - P_1^*) + |\Lambda|^2. \end{aligned}$$

Установим теперь, что

$$(\Lambda, Q - P_1^*) < 0. \quad (1.10)$$

В самом деле,

$$\begin{aligned} (\Lambda, R_\mu) &= (\Lambda, \mu P_1^* + (1 - \mu) P_2^*) = (\Lambda, (\mu - 1)(P_1^* - P_2^*) + P_1^*) = \\ &= (\Lambda, (\mu - 1) \Lambda + P_1^*) = (\mu - 1) |\Lambda|^2 + (\Lambda, P_1^*). \end{aligned}$$

Но $|\Lambda|^2 = \sigma^2 > 0$, $\mu < 1$. Следовательно,

$$(\Lambda, R_\mu) < (\Lambda, P_1^*).$$

Сравнивая полученное соотношение с (1.9), приходим к неравенству

$$(\Lambda, P_1^*) > (\Lambda, Q),$$

эквивалентному (1.10).

Положим

$$(\Lambda, Q - P_1^*) = \alpha \quad (\alpha < 0), \quad |Q - P_1^*| = \beta.$$

Тогда

$$\varphi(\varepsilon) = |Q_\varepsilon - P_2^*|^2 = \beta^2 \varepsilon^2 + 2\varepsilon\alpha + \sigma^2.$$

Очевидно,

$$\varphi(0) = \sigma^2, \quad \varphi'(0) = 2\alpha < 0.$$

Следовательно, при достаточно малом $\varepsilon > 0$

$$\varphi(\varepsilon) < \varphi(0) = \sigma^2.$$

Итак, допустив справедливость неравенства (1.9), приходим к соотношению, противоречащему (1.7). Следовательно, $(\Lambda, Q) > (\Lambda, R_\mu)$ для любой точки $Q \in G_1$. Аналогично доказывается, что $(\Lambda, Q) < (\Lambda, R_\mu)$ для всех точек $Q \in G_2$.

Таким образом, гиперплоскость

$$(\Lambda, X) = c, \quad \text{где } \Lambda = P_1^* - P_2^*, \quad c = (\Lambda, R_\mu),$$

$$R_\mu = \mu P_1^* + (1 - \mu) P_2^*, \quad 0 < \mu < 1,$$

строго разделяет множества G_1 и G_2 .

Теорема полностью доказана.

Если отказаться хотя бы от одного из предположений теоремы 1.1 относительно множеств G_1 и G_2 , то теорема перестанет быть верной. На рис. 3.2 изображен случай, когда G_1 — круг с центром в точке O_1 радиуса R_1 ; G_2 — внутренность круга с центром O_2 радиуса R_2 , причем граничные окружности кругов касаются. Очевидно, G_1 и G_2 не имеют общих точек (так как G_2 не содержит точек граничной окружности), оба множества ограничены. Однако не существует прямой, которая *строго* разделяет множества G_1 и G_2 . Следовательно, отказ от предположения

о замкнутости множеств G_1 и G_2 делает утверждение теоремы 1.1, вообще говоря, неверным.

Пример, иллюстрирующий существенность предположения теоремы 1.1 об ограниченности одного из разделяемых множеств, изображен на рис. 3.3.

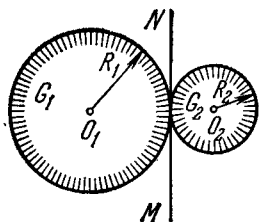


Рис. 3.2.

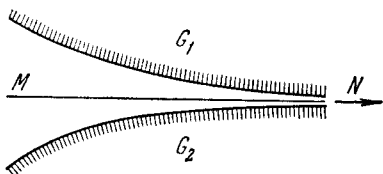


Рис. 3.3.

Иногда нет необходимости требовать, чтобы гиперплоскость строго разделяла два множества. Достаточно иметь гиперплоскость, которая просто разделяет эти множества. В таких случаях может оказаться полезным следующее утверждение.

Теорема 1.2. Если G_1 и G_2 — произвольные выпуклые множества без общих точек, то существует гиперплоскость, разделяющая G_1 и G_2 .

Заметим, что в случаях, изображенных на рис. 3.2 и 3.3, разделяющей прямой является прямая MN .

Если отказаться от предположения о выпуклости каждого из разделяемых множеств, то перестает быть верной не только теорема 1.1, но и теорема 1.2. Соответствующий случай изображен на рис. 3.4.

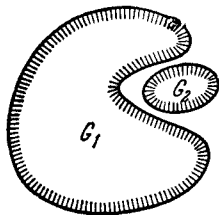


Рис. 3.4.

1.7. В качестве следствия из теоремы о разделяющей гиперплоскости установим два полезных предложения.

Будем говорить, что гиперплоскость Π с уравнением

$$(\Lambda, X) = c$$

является *опорной гиперплоскостью* множества G (опорной для G) в точке $P_0 \in G_1$, если:

а) $(\Lambda, P_0) = c$ (Π содержит точку P_0);

б) $(\Lambda, P) \leq c$ для всех точек $P \in G$, или $(\Lambda, P) \geq c$ при $P \in G$ (множество G лежит в одном из полупространств, порождаемых Π).

Геометрический смысл понятия опорной гиперплоскости легко усмотреть из рис. 3.5. На рис. 3.5,а изображен двумерный случай, когда опорная гиперплоскость Π является прямой линией; на рис. 3.5,б опорная гиперплоскость Π представляет собой обычную плоскость.

Следствие 1.1 (теорема об опорной гиперплоскости). Если P_0 — граничная точка выпуклого замкнутого множества G , то существует опорная гиперплоскость множества G в точке P_0 .

Доказательство. Поскольку P_0 — граничная точка множества G , найдется последовательность $\{P_k\}$, которая состоит из точек, не принадлежащих G , и сходится к P_0 . Рассмотрим выпуклые множества G_1 и G_2 , из которых первым является точка P_k , а второе совпадает с G . Согласно теореме 1.1 существует такая гиперплоскость Π_k с уравнением

$$(\Lambda_k, X) = c_k, \quad (1.11)$$

что

$$(\Lambda_k, P_k) > c_k, \quad (1.12)$$

$$(\Lambda_k, P) < c_k \text{ для } P \in G. \quad (1.13)$$

Не уменьшая общности, можно считать

$$|\Lambda_k| \leq 1, \quad |c_k| \leq 1.$$

Пользуясь теоремой Больцано — Вейерштрасса, выделим из последовательности векторов $\{\Lambda_k\}$ и последовательности чисел $\{c_k\}$ подпоследовательности, сходящиеся соответственно к Λ и c . Покажем, что гиперплоскость $(\Lambda, X) = c$ является искомой.

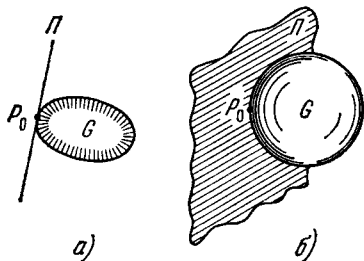


Рис. 3.5.

Действительно, переходя в неравенствах (1.13) к пределу по выделенной подпоследовательности индексов k при фиксированном $P \in G$, получаем

$$(\Lambda, P) \leqslant c \text{ для } P \in G. \quad (1.14)$$

В частности,

$$(\Lambda, P_0) \leqslant c. \quad (1.15)$$

С другой стороны, переход к пределу по указанной подпоследовательности в (1.12) приводит к неравенству

$$(\Lambda, P_0) \geqslant c. \quad (1.16)$$

Сравнивая (1.15) и (1.16), получаем

$$(\Lambda, P_0) = c. \quad (1.17)$$

Соотношения (1.14) и (1.17) показывают, что гиперплоскость $(\Lambda, X) = c$ является опорной для G в точке P_0 . Доказательство следствия закончено.

Отметим, что условия теоремы об опорной гиперплоскости могут быть несколько ослаблены: множество G не обязательно предполагать замкнутым.

Следствие 1.2. Пусть G_1 и G_2 — множества, удовлетворяющие условиям теоремы 1.1. Существует гиперплоскость, разделяющая множества G_1 и G_2 , опорная для G_2 и содержащая G_1 внутри одного из своих полупространств.

Доказательство. Рассмотрим гиперплоскость Π_μ , определяемую уравнением (1.8). В процессе доказательства теоремы 1.1 было установлено, что при $0 < \mu < 1$

$$\min_{P \in G_1} (\Lambda, P) > (\Lambda, R_\mu), \quad (1.18)$$

$$\max_{P \in G_2} (\Lambda, P) < (\Lambda, R_\mu). \quad (1.19)$$

Устремляя параметр μ в (1.19) к нулю, получаем

$$(\Lambda, P) \leqslant (\Lambda, P_2^*),$$

где $P_2^* \in G_2$, а P — произвольная точка G_2 . Следовательно, гиперплоскость Π_0 является опорной для множества G_2 . Очевидно, величина

$$(\Lambda, R_\mu) = \mu |\Lambda|^2 + (\Lambda, P_2^*)$$

убывает с уменьшением μ . Поэтому из (1.18) вытекает, что множество G_1 расположено внутри того полупространства гиперплоскости P_0 , которое не содержит множества G_2 .

Итак, P_0 — искомая гиперплоскость, существование которой утверждается следствием 1.2.

1.8. Рассмотрим пространство трех измерений. Среди выпуклых множеств этого пространства имеются одномерные (отрезок), двумерные (круг) и трехмерные (шар). Характеристическим свойством трехмерной выпуклой фигуры является то, что не существует плоскости, в которую ее можно было бы погрузить. Двумерное выпуклое множество может быть погружено в некоторую плоскость; однако не существует прямой, содержащей это множество. Одномерное выпуклое множество всегда принадлежит некоторой прямой.

Приведенные соображения подсказывают следующее определение размерности выпуклого множества, расположенного в n -мерном пространстве. Будем говорить, что выпуклое множество имеет размерность r , если оно содержится в некотором r -мерном линейном многообразии и не может принадлежать линейному многообразию размерности, меньшей чем r .

Если вспомнить два различных определения линейного многообразия, то определение размерности выпуклого множества можно сформулировать в следующих двух эквивалентных формах:

1. Размерность выпуклого множества $G \subseteq E_n$ равна $r = n - r$, где r — максимальное число линейно независимых гиперплоскостей, общая часть которых содержит G .

2. Размерность выпуклого множества G равна минимальному числу r линейно независимых векторов X_1, X_2, \dots, X_r таких, что совокупность точек

$$X = X_0 + \sum_{i=1}^r \alpha_i X_i$$

(где $X_0 \in G$, а $\alpha_1, \alpha_2, \dots, \alpha_r$ — произвольные числа) содержит G .

Приведем без доказательства одно полезное утверждение, поясняющее смысл понятия размерности выпуклого множества.

Теорема 1.3. *Выпуклое множество G имеет размерность, равную $\rho \geq 1$, в том и только в том случае, если:*

а) G содержится в некотором ρ -мерном линейном многообразии R ;

б) найдутся точка $P \in G$ и число $\varepsilon > 0$ такие, что в пределах ε -окрестности точки P множество G и многообразие R совпадают.

Из сформулированной теоремы, в частности, следует, что размерность n имеют те и только те выпуклые множества $G \subseteq E_n$, которые содержат внутренние точки.

Выпуклым множеством размерности 0 является точка (общая часть n линейно независимых гиперплоскостей).

Одномерное выпуклое множество расположено на некоторой прямой. Отсюда следует, что в предположении замкнутости оно совпадает либо с прямой, либо с лучом, либо с отрезком. Подчеркнем, что введенное здесь определение размерности относится только к выпуклым множествам. Попытка использовать это определение для любых множеств обречена на неудачу, поскольку такое одномерное множество, как окружность, согласно данному определению, является двумерным множеством.

Точку P множества G назовем *крайней*, если не существует двух различных точек P_1 и P_2 , принадлежащих G , таких, что

$$P = \mu P_1 + (1 - \mu) P_2, \quad (1.20)$$

где $0 < \mu < 1$. Таким образом, если P — крайняя точка множества G , а S — произвольный отрезок, принадлежащий G , то P либо не содержится в S , либо является одним из его концов.

Примерами крайних точек линейных и плоских множеств являются концы отрезка, вершины многоугольника, точки граничной окружности круга и т. д. Выпуклые множества могут иметь как конечное (многоугольник), так и бесконечное (круг) число крайних точек. Некоторые выпуклые множества (например, прямая) не содержат ни одной крайней точки. Ниже мы увидим, что подобные

множества обязаны быть либо неограниченными, либо незамкнутыми.

На рис. 3.6 изображены три выпуклых множества: четырехугольник G_1 , круг G_2 , прямая G_3 . Ни одна из вершин четырехугольника G_1 не может быть погружена внутрь отрезка, целиком содержащегося в G_1 . Следовательно, A, B, C и D — крайние точки четырехугольника G_1 . Любая другая точка G_1 не является крайней. (Например, S лежит внутри отрезка $[C, B] \subseteq G_1$.)

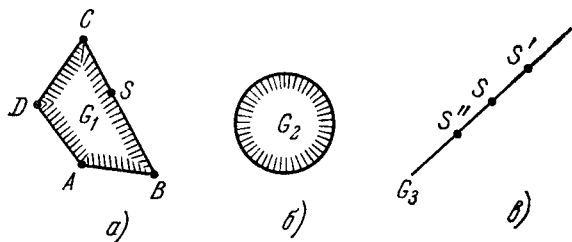


Рис. 3.6.

Геометрически очевидно, что все точки граничной окружности круга G_2 (рис. 3.6, б) являются крайними. Прямая (рис. 3.6, в) не имеет ни одной крайней точки, так как произвольная ее точка S может быть помещена внутри отрезка $[S', S''] \subseteq G_3$.

Будем говорить, что множество G является *выпуклой оболочкой* множества G_1 , если оно состоит из всевозможных точек вида

$$\sum_{i=1}^N \mu_i P_i,$$

где

$$\sum_{i=1}^N \mu_i = 1, \quad \mu_i \geq 0 \quad (i = 1, 2, \dots, N),$$

P_1, P_2, \dots, P_N — произвольная конечная система точек из G_1 .

Вектор $\sum_{i=1}^N \mu_i P_i$ принято называть *выпуклой комбинацией* векторов P_1, P_2, \dots, P_N . Нетрудно заметить, что

выпуклая оболочка произвольного множества есть выпуклое множество. С другой стороны, любое выпуклое множество, содержащее G_1 , обязано включать его выпуклую оболочку G .

Поэтому выпуклая оболочка множества G — минимальное выпуклое множество, содержащее G .

Приведем без доказательства утверждение, которое дает возможность выяснить структуру выпуклого замкнутого ограниченного множества.

Теорема 1.4 (теорема о представлении). Пусть G — выпуклое замкнутое ограниченное множество, G^* — совокупность крайних точек G . В таком случае G является выпуклой оболочкой множества G^* .

Пусть G — произвольное выпуклое множество. Множество G_0 назовем *остовом* G , если:

- а) G — выпуклая оболочка G_0 ;
- б) любое подмножество G_0 не удовлетворяет условию а).

Образно говоря, остов множества является наиболее экономным множеством из числа удовлетворяющих условию а).

Заметим, что не любое выпуклое множество обладает остовом. Например, открытый интервал остова не имеет. Теорема о представлении устанавливает существование остова для произвольного выпуклого замкнутого ограниченного множества. При этом оказывается, что остов множества G , удовлетворяющего перечисленным условиям, совпадает с G^* — множеством крайних точек G .

В заключение приведем одно очевидное следствие теоремы 1.4.

Следствие 1.3. Всякое непустое выпуклое замкнутое ограниченное множество содержит хотя бы одну крайнюю точку.

§ 2. ВЫПУКЛЫЕ МНОГОГРАННЫЕ МНОЖЕСТВА

2.1. Рассмотрим совокупность планов произвольной задачи линейного программирования с n переменными. Другими словами, введем в рассмотрение множество точек (векторов) $X = (x_1, x_2, \dots, x_n)$, определяемое произвольной системой линейных ограничений (равенств и

неравенств):

$$(D_i, X) = \sum_{j=1}^n d_{ij}x_j \leq d_i, \quad i=1, 2, \dots, s, \quad (2.1)$$

$$(D_i, X) = \sum_{j=1}^n d_{ij}x_j = d_i, \quad i=s+1, s+2, \dots, s+t. \quad (2.2)$$

Здесь $D_i = (d_{i1}, d_{i2}, \dots, d_{in})$, $i=1, 2, \dots, s+t$.

Мы не выделяем особо условий вида $x_j \geq 0$, как это часто делается при записи ограничений задачи линейного программирования. Эти условия (если они имеются) включаются в общую систему неравенств (2.1). Обозначим через M множество точек X , удовлетворяющих соотношениям (2.1), (2.2). Будем предполагать, что система ограничений (2.1), (2.2) непротиворечива, т. е. множество M не является пустым множеством.

В этом параграфе отмечается ряд полезных и важных свойств множества M , на которых в значительной степени основана теория линейного программирования.

Теорема 2.1. *Множество M является выпуклым замкнутым множеством.*

Доказательство. 1. Пусть X_1, X_2 — произвольные точки M ; $0 \leq \alpha \leq 1$. В таком случае точка $X = \alpha X_1 + (1-\alpha) X_2 \in M$. Действительно,

$$(D_i, X) = \alpha (D_i, X_1) + (1-\alpha) (D_i, X_2).$$

Следовательно,

$$(D_i, X) \leq \alpha d_i + (1-\alpha) d_i = d_i \quad \text{для } i=1, 2, \dots, s,$$

$$(D_i, X) = \alpha d_i + (1-\alpha) d_i = d_i \quad \text{для } i=s+1, s+2, \dots, s+t,$$

т. е. $X \in M$. Согласно определению это означает, что M — выпуклое множество.

2. Пусть X_k , $k=1, 2, \dots$ — произвольная последовательность точек из M , сходящаяся к точке X . Учитывая свойство непрерывности скалярного произведения векторов, получаем

$$\lim_{k \rightarrow \infty} (D_i, X_k) = (D_i, X) \quad \text{при любом } i. \quad (2.3)$$

По условию

$$\begin{aligned}(D_i, X_k) &\leq d_i \quad \text{для } i = 1, 2, \dots, s, \\ (D_i, X_k) &= d_i \quad \text{для } i = s+1, s+2, \dots, s+t \\ &\quad (k = 1, 2, \dots).\end{aligned}$$

Переходя в этих соотношениях к пределу при $k \rightarrow \infty$ и учитывая (2.3), получаем, что точка X удовлетворяет системе условий (2.1), (2.2), т. е. принадлежит множеству M . Таким образом, множество M содержит все свои предельные точки и, следовательно, является замкнутым множеством. Теорема доказана.

Структура ограничений (2.1), (2.2) показывает, что множество M является общей частью полупространств

$$(D_i, X) \leq d_i, \quad i = 1, 2, \dots, s,$$

и гиперплоскостей

$$(D_i, X) = d_i, \quad i = s+1, s+2, \dots, s+t,$$

n -мерного пространства точек $X = (x_1, x_2, \dots, x_n)$.

Множество, образованное пересечением конечного числа полупространств и гиперплоскостей (если это пересечение не пусто), будем называть *выпуклым многогранным множеством*. Таким образом, M — выпуклое многогранное множество. Если, в частности, $s=0$, т. е. ограничения (2.1) отсутствуют, то M — общая часть гиперплоскостей с уравнениями (2.2) — представляет собой линейное многообразие. Таким образом, понятие выпуклого многогранного множества может рассматриваться как обобщение понятия линейного многообразия. Выпуклое многогранное множество является геометрическим образом совместной системы, состоящей из конечного числа линейных равенств и неравенств. Поскольку в дальнейшем речь будет идти только о выпуклых многогранных множествах, слово «выпуклое» в тех местах, где это не может привести к недоразумению, будет опускаться.

Систему ограничений (2.1), (2.2), порождающую выпуклое многогранное множество M , будем называть иногда *системой условий* этого множества.

Выпуклым многогранником называется ограниченное выпуклое многогранное множество. Таким образом, сис-

тема соотношений (2.1), (2.2) в случае ограниченности множества M определяет выпуклый многогранник n -мерного пространства. В соответствии с введенными определениями совокупность планов задачи линейного программирования естественно называть *многогранным множеством* (*многогранником*) условий задачи.

2.2. В п. 1.8 было введено понятие размерности выпуклого множества. Выясним связь между размерностью многогранного множества M и свойствами условий (2.1), (2.2), определяющих это множество. Предварительно приведем несколько определений.

Условие с номером i системы (2.1), (2.2) назовем *жестким ограничением* многогранного множества M , если любая точка M удовлетворяет ему как точному равенству, т. е.

$$(D_i, X) = d_i \quad \text{при любом } X \in M.$$

Очевидно, любое из условий (2.2) является жестким ограничением M .

Жесткими ограничениями могут оказаться также и некоторые из условий системы (2.1).

Для примера рассмотрим многогранное множество M' , определяемое системой неравенств

$$\left. \begin{aligned} x_1 + x_2 &\leq 1, \\ 2x_1 + x_2 &\leq 2, \\ -3x_1 - 2x_2 &\leq -3, \\ x_3 &\leq 1. \end{aligned} \right\} \quad (2.4)$$

Пусть точка $X = (x_1, x_2, x_3) \in M'$. Складывая первые два условия системы (2.4) и сравнивая результат с третьим неравенством этой системы, получаем

$$-3x_1 - 2x_2 = -3.$$

Таким образом, третье неравенство системы (2.4) является жестким ограничением многогранного множества M' .

Условие с номером i системы, определяющей многогранное множество M , назовем *нежестким ограничением* этого множества, если существует такая точка $X \in M$, что

$$(D_i, X) < d_i.$$

Итак, система условий, определяющая многогранное множество, может быть разделена на две подсистемы, из которых первая состоит из жестких, а вторая — из нежестких ограничений данного множества.

Очевидно, нежесткими ограничениями могут оказаться лишь условия-неравенства. В приведенном выше примере четвертое неравенство — нежесткое ограничение M' , так как точка $(x_1, x_2, x_3) = (1, 0, 0) \in M'$ и $x_3 = 0 < 1$. Нетрудно проверить, что первые два неравенства системы (2.4) являются жесткими ограничениями множества M' .

Поясним геометрический смысл введенных определений. Неравенство

$$(D_i, X) \leq d_i$$

высекает в пространстве точек X полупространство P_i с граничной гиперплоскостью Γ_i , определяемой уравнением

$$(D_i, X) = d_i.$$

Жесткость i -го условия ($i = 1, 2, \dots, s$) системы (2.1) означает, что множество M , которое обязано содержаться в полупространстве P_i , принадлежит на самом деле его границе Γ_i . Если же i -е условие нежесткое, то M выходит за пределы гиперплоскости Γ_i : некоторые точки множества M являются внутренними точками полупространства P_i .

Систему линейных ограничений, состоящую из условий типа (2.1), (2.2), будем называть *линейно-независимой*, если соответствующая система векторов D_i линейно независима. Рангом системы линейных ограничений назовем ранг матрицы, составленной из векторов D_i , отвечающих данной системе.

Приводимое ниже утверждение устанавливает связь между размерностью множества и рангом системы его жестких ограничений.

Теорема 2.2. *Размерность ρ многогранного множества M определяется формулой*

$$\rho = n - \sigma, \quad (2.5)$$

где σ — ранг системы жестких ограничений этого множества.

Обратимся к многогранному множеству M' , задаваемому системой ограничений (2.4). Жесткими ограничениями являются первые три неравенства системы (2.4).

Первые два ограничения линейно независимы. Третье — их линейная комбинация. Следовательно, в данном случае $\sigma = 2$. Поэтому размерность многогранного множества M' равна

$$\rho = n - \sigma = 3 - 2 = 1.$$

Система условий, определяющая многогранное множество, может быть приведена к эквивалентному виду, в котором каждое неравенство является нежестким ограничением, а все условия-равенства линейно независимы. Для этого достаточно заменить в каждом жестком ограничении-неравенстве знак неравенства на знак равенства, а затем исключить из полученной системы равенств те уравнения, которые являются следствиями остальных. Полученная система условий эквивалентна исходной системе в том смысле, что каждая из них определяет одно и то же многогранное множество. Если предположить, что указанные преобразования уже проведены и условия (2.1), (2.2) — полученная таким образом система ограничений, то в соответствии с теоремой 2.2 размерность ρ множества M определяется формулой

$$\rho = n - t. \quad (2.6)$$

2.3. Для дальнейшего нам понадобится понятие грани произвольного многогранного множества. Подмножество G многогранного множества M будем называть q -мерной гранью M , если:

а) размерность G равна q ;

б) из условий $X = \alpha X' + (1 - \alpha) X'' \in G$, $0 < \alpha < 1$ и $X', X'' \in M$ следует, что $X', X'' \in G$.

Как нетрудно видеть, при $q = 0$ приведенное определение превращается в определение крайней точки множества M . Крайние точки многогранного множества естественно называть *вершинами* этого множества. Таким образом, 0-мерная грань M и вершина M — понятия эквивалентные.

Под *ребром* многогранного множества будем понимать произвольную одномерную грань этого множества.

Определение грани многогранного множества M может быть также дано в терминах, связанных с системой условий (2.1), (2.2). Именно, q -мерной гранью множества M является произвольное q -мерное многогранное множество, система условий которого образуется из (2.1), (2.2) путем замены некоторых знаков неравенства знаками равенства. Отметим, что в силу теоремы 2.2 число линейно независимых жестких ограничений q -мерной грани равно $n - q$.

Справедливо следующее утверждение.

Теорема 2.3. Оба приведенных выше определения q -мерной грани многогранного множества эквивалентны.

Очевидно, каждая грань многогранного множества M однозначно определяется системой своих жестких ограничений. Отсюда непосредственно следует, что число q -мерных граней многогранного множества M не превышает общего количества наборов по $n - q$ линейно независимых ограничений системы (2.1), (2.2), которое в свою очередь не более C_{n-q}^{n-q} . Полученную оценку для числа q -мерных граней M без труда можно было бы уточнить. Однако нам важно лишь то обстоятельство, что любое многогранное множество имеет конечное число граней.

Ниже будет отмечено, что для воспроизведения большинства многогранных множеств достаточно располагать их вершинами и ребрами. Поэтому естественно остановиться несколько подробнее на этих простейших гранях выпуклых многогранных множеств.

В качестве непосредственного следствия из теоремы 2.3 укажем важное для дальнейшего характеристическое свойство вершин произвольного многогранного множества.

Теорема 2.4. Точка $X \in M$ является вершиной многогранного множества M в том и только в том случае, если среди условий (2.1), (2.2) найдутся n линейно независимых ограничений, которым эта точка удовлетворяет как точным равенствам.

Для доказательства теоремы достаточно вспомнить, что вершина многогранного множества и его грань нулевой размерности — понятия эквивалентные, а затем использовать второе определение грани многогранного множества и теорему 2.2.

Займемся теперь анализом ребер многогранного множества (его одномерных граней).

Пусть Γ — некоторое ребро многогранного множества M . Будучи одномерным выпуклым множеством, Γ является либо отрезком, либо лучом, либо, наконец, прямой. Поэтому множество точек X , составляющих ребро Γ , может быть представлено в виде

$$X = X_0 + \lambda l. \quad (2.7)$$

Здесь X_0 — некоторая точка ребра Γ ; l — направляющий вектор этого ребра; λ — параметр, пределы изменения которого определяют вид данного ребра.

Если Γ — отрезок, луч или прямая, то параметр λ изменяется соответственно в пределах

$$\lambda' \leq \lambda \leq \lambda'', \quad (2.8)$$

$$\lambda' \leq \lambda < \infty, \quad (2.9)$$

$$-\infty < \lambda < \infty. \quad (2.10)$$

Выделим все жесткие ограничения Γ и их номера объединим в множество E_Γ . По условию среди уравнений

$$(D_i, X) = d_i, \quad i \in E_\Gamma, \quad (2.11)$$

имеется $n-1$ линейно независимых.

Пусть $X' = X_0 + \lambda' l$ и $X'' = X_0 + \lambda'' l$ — две различные точки Γ . Учитывая, что при любом $i \in E_\Gamma$ точки X' и X'' удовлетворяют i -му уравнению (2.11), имеем

$$(D_i, l) = \left(D_i, \frac{X' - X''}{\lambda' - \lambda''} \right) = \frac{1}{\lambda' - \lambda''} (d_i - d_i) = 0 \quad \text{для } i \in E_\Gamma.$$

Таким образом, направляющий вектор l ребра Γ является ненулевым решением системы однородных уравнений

$$(D_i, X) = 0, \quad i \in E_\Gamma, \quad (2.12)$$

матрица которой имеет ранг, равный $(n-1)$.

Для того чтобы выяснить вид ребра Γ , необходимо вычислить числа (D_i, l) для $i \notin E_\Gamma$. Очевидно, наличие границ (2.8), (2.9) и (2.10) эквивалентно соответственно условиям: среди чисел (D_i, l) имеются как положительные

так и отрицательные; среди чисел (D_i, l) имеются отрицательные, однако нет положительных; все числа (D_i, l) равны нулю. Заметим, что последнее условие означает наличие у системы уравнений

$$(D_i, X) = 0, \quad i = 1, 2, \dots, s + t,$$

ненулевого решения. Следовательно, ранг системы условий многогранного множества, имеющего в качестве ребра прямую линию, должен быть меньше n .

Рассмотрим концы ребра Γ , т. е. точки $X(\lambda')$, $X(\lambda'')$, если Γ — отрезок, и точку $X(\lambda')$, если Γ — луч. Как нетрудно проверить, эти точки являются вершинами многогранного множества M .

Резюмируя все сказанное относительно ребер многогранного множества, приходим к следующему предложению.

Теорема 2.5. *Ребро Γ многогранного множества M представляет собой либо отрезок, соединяющий две вершины M , либо луч, исходящий из вершины M , либо прямую линию. При этом направляющий вектор l ребра Γ удовлетворяет системе однородных уравнений (2.12), среди которых имеется $(n-1)$ линейно независимых уравнений. Ребро Γ — отрезок, если среди чисел (D_i, l) имеются как положительные, так и отрицательные; ребро Γ — луч, если среди этих чисел нет положительных, но есть отрицательные; ребро Γ — прямая, если все числа $(D_i, l) = 0$.*

2.4. В п. 1.8 была приведена теорема о представлении, согласно которой произвольная точка выпуклого ограниченного замкнутого множества является выпуклой линейной комбинацией крайних точек этого множества. Применяя отмеченный результат к ограниченному многогранному множеству (выпуклому многограннику) и учитывая, что число его крайних точек — вершин — конечно, приходим к следующему утверждению.

Теорема 2.6. *Произвольный выпуклый многогранник, определяемый системой условий (2.1), (2.2), совпадает с совокупностью точек X вида*

$$X = \sum_{i=1}^N \alpha_i X_i, \quad (2.13)$$

где X_i , $i = 1, 2, \dots, N$, — вершины многогранника;

$$\sum_{i=1}^N \alpha_i = 1; \text{ все } \alpha_i \geq 0.$$

Таким образом, теорема о представлении полностью определяет структуру выпуклого многогранника и выявляет особую роль его вершин. Однако для всякого неограниченного многогранного множества утверждение теоремы о представлении заведомо неверно. Действительно, пусть M — неограниченное многогранное множество. Поскольку M обладает конечным числом вершин, то множество точек, представимых в виде (2.13), является ограниченным. Следовательно, оно не может совпадать с M . Таким образом, формула (2.13) не может служить для представления неограниченного многогранного множества. Этот и следующий пункты посвящены теореме о представлении для произвольного выпуклого многогранного множества.

Предположим вначале, что ранг системы ограничений (2.1), (2.2), определяющей многогранное множество M , равен n — числу компонент вектора X . Этот случай особенно важен, поскольку, как будет показано в § 3, любая задача линейного программирования может быть приведена к эквивалентной задаче, многогранное множество условий которой обладает отмеченным свойством. Приводимая ниже без доказательства теорема указывает на возможность восстановления многогранного множества по его вершинам и направляющим векторам неограниченных ребер.

Теорема 2.7. Пусть ранг совместной системы (2.1), (2.2) равен n . В таком случае выпуклое многогранное множество M совпадает с совокупностью точек X вида

$$X = \sum_{i=1}^{N_1} \alpha_i X_i + \sum_{i=1}^{N_2} \beta_i R_i. \quad (2.14)$$

Здесь X_i , $i = 1, 2, \dots, N_1$, — полный набор вершин M ; R_i , $i = 1, 2, \dots, N_2$, — полный набор направляющих векторов неограниченных ребер M ; все числа α_i , β_i неотрицательны; $\sum_{i=1}^{N_1} \alpha_i = 1$.

Теорема 2.7 содержит в качестве частного случая результат теоремы 2.6. Действительно, если M — выпуклый многогранник, то M не имеет неограниченных ребер, т. е. $N_2 = 0$. В этом случае представление (2.14) переходит в представление (2.13), фигурирующее в теореме 2.6.

В качестве следствия из теоремы 2.7 получаем одно важное предложение, сформулированное ниже в виде теоремы.

Теорема 2.8. Если ранг совместной системы ограничений (2.1), (2.2), определяющей многогранное множество M , равен n , то M обладает хотя бы одной вершиной.

Для доказательства теоремы 2.8 достаточно представить произвольную точку множества M в виде (2.14).

Согласно теореме 2.4 каждой вершине многогранного множества M отвечает n линейно независимых ограничений системы (2.1), (2.2), которым эта точка удовлетворяет как равенствам. Поэтому наличие у M хотя бы одной вершины означает, что ранг системы (2.1), (2.2) равен n . Объединяя этот вывод с теоремой 2.8, приходим к следующему результату.

Теорема 2.9. Для того чтобы непустое многогранное множество M имело вершины, необходимо и достаточно, чтобы ранг системы ограничений (2.1), (2.2) был равен n .

2.5. Теорема 2.7 устанавливает структуру многогранного множества M для наиболее важного случая, когда ранг системы условий (2.1), (2.2) в точности равен n — размерности пространства точек X . Исследуем теперь общий случай, в котором ранг системы ограничений (2.1), (2.2) равен $r \leq n$.

Итак, пусть ранг системы векторов D_i ($i=1, 2, \dots, s+t$) равен $r < n$. Это означает, что в матрице $\|d_{ij}\|_{s+t,n}$ имеется r линейно независимых столбцов, причем остальные $n-r$ столбцов матрицы являются их линейными комбинациями. Обозначим j -й столбец матрицы $\|d_{ij}\|_{s+t,n}$ через

$$D^{(j)} = (d_{1j}, d_{2j}, \dots, d_{s+t,j})^T, \quad j=1, 2, \dots, n. \quad (2.15)$$

Тогда система условий (2.1), (2.2) может быть записана в виде

$$\sum_{j=1}^n x_j D^{(j)} \leqslant D. \tag{2.16}$$

Здесь $D = (d_1, d_2, \dots, d_{s+t})^T$; знак \leqslant указывает, что первые s составляющих векторного соотношения (2.16) являются неравенствами, а последующие t составляющих — равенствами.

По условию максимальная линейно независимая подсистема системы векторов $D^{(j)}$, $j = 1, 2, \dots, n$, состоит из r векторов. Без ограничения общности можно принять, что этими векторами являются первые r векторов системы (2.15)

$$D^{(1)}, D^{(2)}, \dots, D^{(r)}. \tag{2.17}$$

Выразим векторы $D^{(j)}$ при $j > r$ в виде линейных комбинаций векторов (2.17)

$$D^{(j)} = \sum_{i=1}^r \delta_{ij} D^{(i)}. \tag{2.18}$$

С помощью равенств (2.18) система ограничений (2.16) легко приводится к эквивалентной форме:

$$\sum_{i=1}^r \left(x_i + \sum_{j=r+1}^n x_j \delta_{ij} \right) D^{(i)} \leqslant D. \tag{2.19}$$

Рассмотрим многогранное множество M_0 , система условий которого состоит из (2.16) (или (2.19)) и уравнений

$$x_{r+1} = x_{r+2} = \dots = x_n = 0. \tag{2.20}$$

Как нетрудно видеть, ранг системы ограничений многогранного множества M_0 равен в точности n .

Приводимое ниже без доказательства утверждение устанавливает связь между многогранными множествами M и M_0 , которая позволит использовать для анализа структуры M результат теоремы 2.7.

Обозначим через Π_{n-r} $(n-r)$ -мерное подпространство, которое образовано пересечением r $(n-1)$ -мерных

подпространств (гиперплоскостей), задаваемых линейно независимыми однородными уравнениями

$$L_i(X) = x_i + \sum_{j=r+1}^n \delta_{ij} x_j = 0, \quad i = 1, 2, \dots, r. \quad (2.21)$$

Лемма 2.1. *Многогранное множество M является суммой*) многогранного множества M_0 и подпространства Π_{n-r} :*

$$M = M_0 + \Pi_{n-r}.$$

Из леммы 2.1, в частности, следует, что равенство $r = n$ является необходимым условием ограниченности многогранного множества M .

Утверждение леммы 2.1 позволяет также перенести теорему 2.7 на случай произвольного многогранного множества.

Теорема 2.10 (теорема о представлении многогранного множества). *Произвольное многогранное множество M совпадает с совокупностью точек X вида*

$$X = \sum_{i=1}^{N_1} \alpha_i X_i + \sum_{i=1}^{N_2} \beta_i R_i + \sum_{i=1}^{n-r} \gamma_i R'_i, \quad (2.22)$$

где $\sum_{i=1}^{N_1} \alpha_i = 1$, $\alpha_i \geq 0$, $\beta_i \geq 0$; X_i — вершины M_0 ; R_i — направляющие векторы неограниченных ребер M_0 ; R'_i , $i = 1, 2, \dots, n-r$, — любая полная система линейно независимых векторов, содержащихся в пространстве Π_{n-r} ; r — ранг системы ограничений, определяющей M .

Теорема 2.10 является очевидным следствием теоремы 2.7 и леммы 2.1.

Теорема 2.10 устанавливает вид общего решения произвольной системы линейных равенств и неравенств и представляет собой естественное обобщение соответствующей теоремы теории линейных уравнений.

Для иллюстрации результатов настоящего параграфа приведем два численных примера.

*) По определению множество A является суммой множеств B и C ($A = B + C$), если условие $X \in A$ эквивалентно представлению $X = Y + Z$, где $Y \in B$, $Z \in C$.

Пример 1. Рассмотрим многогранное множество M_1 , определяемое системой условий

$$\left. \begin{aligned} -x_1 - x_2 &\leq 0, \\ -x_1 + x_2 &\leq 0, \\ -x_2 &\leq 2, \\ -x_1 + 2x_2 &\leq 2. \end{aligned} \right\} \quad (2.23)$$

Изображение M_1 в плоскости x_1Ox_2 приведено на рис. 3.7. Из рис. 3.7 видно, что границами M_1 минимальной (нулевой) размерности являются точки $X_1=(2, -2)$, $X_2=(2, 2)$, $X_3=(0, 0)$. Это обстоятельство находится в соответствии с теоремой 2.9, поскольку ранг системы (2.23) равен $n=2$ (векторы $D_1=(-1, -1)$, $D_2=(-1, 1)$ линейно независимы). Многогранное множество M_1 не ограничено и, следовательно, должно иметь неограниченные ребра. Из рис. 3.7 видно, что у этого множества два неограниченных ребра, которые исходят из точек X_1 и X_2 и имеют в качестве направляющих векторов $R_1=(1, 0)$ и $R_2=(2, 1)$.

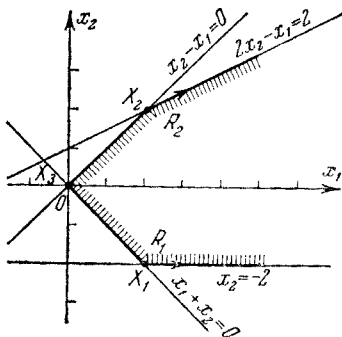


Рис. 3.7.

Согласно теореме 2.7 многогранное множество M_1 может быть представлено в виде суммы произвольной выпуклой комбинации точек X_1 , X_2 , X_3 и произвольной неотрицательной комбинации векторов R_1 и R_2 , т. е. M_1 совпадает с совокупностью точек

$$X = \alpha_1(2, -2) + \alpha_2(2, 2) + \alpha_3(0, 0) + \beta_1(1, 0) + \beta_2(2, 1),$$

где $\alpha_i \geq 0$, $\beta_i \geq 0$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$, или

$$X = (2\alpha_1 + 2\alpha_2 + \beta_1 + 2\beta_2, -2\alpha_1 + 2\alpha_2 + \beta_2),$$

где $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\beta_1 \geq 0$, $\beta_2 \geq 0$, $\alpha_1 + \alpha_2 \leq 1$.

Пример 2. Рассмотрим многогранное множество M , определяемое системой ограничений

$$\left. \begin{aligned} -x_1 - x_2 - 2x_3 &\leq 0, \\ -x_1 + x_2 &\leq 0, \\ -x_2 - x_3 &\leq 2, \\ -x_1 + 2x_2 + x_3 &\leq 2. \end{aligned} \right\} \quad (2.24)$$

В данном случае $n=3$, однако, как легко заметить, ранг системы (2.24) равен 2.

Положим

$$D^{(1)} = (-1, -1, 0, -1)^T,$$

$$D^{(2)} = (-1, 1, -1, 2)^T,$$

$$D^{(3)} = (-2, 0, -1, 1)^T.$$

Очевидно, векторы $D^{(1)}$ и $D^{(2)}$ линейно независимы, а $D^{(3)} = D^{(1)} + D^{(2)}$. Многогранное множество M_0 в соответствии с общим правилом определяется условиями (2.24) и дополнительным равенством $x_3 = 0$.

Мы видим, что M_0 совпадает с многогранным множеством M_1 , рассмотренным в предыдущем примере. Поскольку в данном случае $n-r=1$, подпространство $\Pi_{n-r} = \Pi_1$ является прямой. Преобразовав условия (2.24) к виду

$$-(x_1 + x_3) - (x_2 + x_3) \leq 0,$$

$$-(x_1 + x_3) + (x_2 + x_3) \leq 0,$$

$$-(x_1 + x_3) - (x_2 + x_3) \leq 2,$$

$$-(x_1 + x_3) + 2(x_2 + x_3) \leq 2,$$

получаем, что эта прямая определяется соотношениями

$$\left. \begin{aligned} x_1 + x_3 &= 0, \\ x_2 + x_3 &= 0. \end{aligned} \right\} \quad (2.25)$$

В соответствии с леммой 2.1 многогранное множество M может рассматриваться как совокупность прямых, параллельных (2.25) и проходящих через точки многоугольного множества M_0 . Изображение M дано на рис. 3.8.

Минимальная размерность граней M равна 1.

Многогранное множество M имеет три грани минимальной размерности — прямые $X_1'X_1''$, $X_2'X_2''$, $X_3'X_3''$. Каждая из этих прямых соответствует определенной вершине M_0 . Выпишем, основываясь на формуле (2.22), общее представление точек $X \in M$.

В данном случае $M_0 = M_1$. Поэтому $X_1 = (2, -2, 0)$, $X_2 = (2, 2, 0)$, $X_3 = (0, 0, 0)$, $R_1 = (1, 0, 0)$, $R_2 = (2, 1, 0)$. Вектор R_1' (единственный, так как $n-r=1$) является направляющим вектором прямой (2.25)

$$R_1' = (-1, -1, 1).$$

Таким образом, многогранное множество M представляет собой совокупность точек X вида

$$X = \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \beta_1 R_1 + \beta_2 R_2 + \gamma_1 R_1'.$$

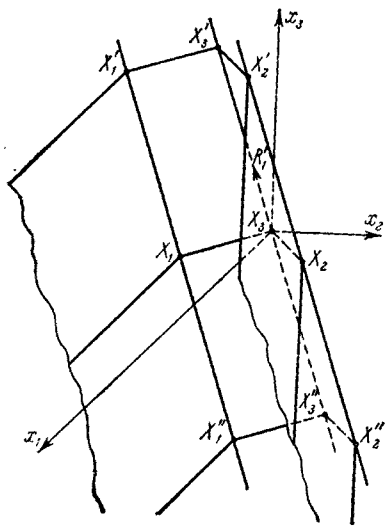


Рис. 3.8.

или

$$X = (2\alpha_1 + 2\alpha_2 + \beta_1 + 2\beta_2 - \gamma_1, -2\alpha_1 + 2\alpha_2 + \beta_2 - \gamma_1, \gamma_1),$$

где $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\beta_1 \geq 0$, $\beta_2 \geq 0$, $\alpha_1 + \alpha_2 \leq 1$.

2.6. Теорему 2.7, устанавливающую представление (2.14) для произвольного многогранного множества при $r=n$, можно переформулировать следующим образом. *Любое многогранное множество M при $r=n$ может быть представлено как сумма множества выпуклых комбинаций одной группы точек (вершин M) и множества неотрицательных линейных комбинаций другой группы точек (направляющих векторов неограниченных ребер M).*

Нетрудно заметить, что любое s -мерное подпространство является множеством неотрицательных линейных комбинаций конечного числа точек. Действительно, если X_1, X_2, \dots, X_s — произвольная линейно независимая система векторов этого подпространства, то в качестве искомой системы может быть принята совокупность точек вида

$$X_1, X_2, \dots, X_s; -X_1, -X_2, \dots, -X_s.$$

Теперь мы в состоянии переформулировать общую теорему о представлении многогранного множества (теорема 2.10) подобно тому, как это было сделано для теоремы 2.7.

Произвольное многогранное множество может быть представлено как сумма множеств выпуклых и неотрицательных линейных комбинаций некоторых систем точек.

В этом пункте приводится обратное утверждение, которое позволит ввести новое определение многогранного множества, полезное в ряде теоретических вопросов линейного программирования.

Пусть X_i , $i=1, 2, \dots, N_1$; R_i , $i=1, 2, \dots, N_2$, — произвольные точки n -мерного пространства. Рассмотрим множество T , состоящее из точек X , представимых в виде

$$X = \sum_{i=1}^{N_1} \alpha_i X_i + \sum_{i=1}^{N_2} \beta_i R_i, \quad (2.26)$$

где $\alpha_i \geq 0$, $\beta_i \geq 0$, $\sum_{i=1}^{N_1} \alpha_i = 1$.

Имеет место следующее утверждение.

Теорема 2.11. *Множество T , состоящее из точек X , представимых в виде (2.26), является выпуклым многогранным множеством.*

2.7. Будем для краткости говорить, что множество T , состоящее из точек X вида (2.26), образовано (или порождается) точками X_1, X_2, \dots, X_{N_1} и векторами R_1, R_2, \dots, R_{N_2} . Напомним, что понятия точки и вектора эквивалентны. Называя X_1, \dots, X_{N_1} точками, а R_1, R_2, \dots, R_{N_2} векторами, мы подчеркиваем лишь различную роль, которую эти элементы играют в образовании множества T : T совпадает с суммой множества *выпуклых* комбинаций точек X_1, X_2, \dots, X_{N_1} и множества *неотрицательных* линейных комбинаций векторов R_1, R_2, \dots, R_{N_2} . Как уже отмечалось, любое многогранное множество порождается конечным числом точек и векторов. С другой стороны, в соответствии с теоремой 2.11 произвольное множество, образованное конечным числом точек и векторов, является многогранным. Это позволяет ввести еще одно определение выпуклого многогранного множества.

Выпуклым многогранным множеством называется множество, порожденное конечным числом точек и векторов (являющееся суммой множества выпуклых комбинаций данных точек и множества неотрицательных линейных комбинаций данных векторов). Сформулированное определение назовем *вторым определением* многогранного множества. Определение, приведенное в п. 2.1, согласно которому многогранное множество является пересечением конечного числа полупространств и гиперплоскостей, будем в дальнейшем именовать *первым определением* многогранного множества. Эквивалентность этих двух определений, устанавливаемая теоремами 2.10 и 2.11, является центральным пунктом теории выпуклых многогранных множеств.

Линейное многообразие является простейшим частным случаем выпуклого многогранного множества, когда система его условий состоит из одних только уравнений. Поэтому приведенные здесь два эквивалентных определения многогранного множества являются обобщением двух различных определений линейного многообразия, экви-

валентность которых доказывается в теории линейных уравнений.

Выше был выделен важный класс выпуклых многогранных множеств: класс выпуклых многогранников. В соответствии с первым определением выпуклый многогранник — это ограниченное множество, являющееся пересечением конечного числа полупространств и гиперплоскостей. Поскольку ограниченное множество не может содержать лучей, то второе определение выпуклого многогранника формулируется следующим образом.

Выпуклым многогранником называется множество, состоящее из всевозможных выпуклых комбинаций конечного числа точек.

Выделим еще один важный класс выпуклых многогранных множеств. В п. 1.5 было дано определение выпуклого конуса, расположенного в произвольном конечномерном пространстве.

Многогранным выпуклым конусом назовем выпуклое многогранное множество, обладающее свойствами конуса. Другими словами, многогранное множество M , определяемое ограничениями (2.1), (2.2), является многогранным конусом с вершиной в точке X_0 (слово выпуклый мы для краткости опускаем), если из условия $X \in M$ вытекает, что

$$X_0 + \lambda(X - X_0) \in M$$

при любом $\lambda \geq 0$.

Нетрудно проверить, что система условий произвольного многогранного конуса может быть приведена к виду

$$(D_i, X) \leq (D_i, X_0), \quad i = 1, 2, \dots, s, \quad (2.27)$$

$$(D_i, X) = (D_i, X_0), \quad i = s+1, \dots, s+t. \quad (2.28)$$

С другой стороны, легко видеть, что любое множество, определяемое условиями (2.27), (2.28), является многогранным конусом. Условия (2.27) и (2.28) позволяют определить выпуклый многогранный конус как общую часть конечного числа полупространств и гиперплоскостей, пересечение граничных гиперплоскостей которых содержит по крайней мере одну точку (граничной гиперплоскостью для гиперплоскости считается она сама).

Обозначим многогранный конус, задаваемый системой ограничений (2.27), (2.28), через K .

Отметим, что ни одна из точек X конуса K , отличная от его вершины X_0 , не может быть крайней точкой K . Действительно вместе с точкой X конус K содержит луч, исходящий из X_0 и проходящий через X . Поэтому $X \neq X_0$ может быть представлена в виде выпуклой комбинации некоторых точек X' и X'' , лежащих на луче по обе стороны от точки X .

Допустим вначале, что ранг системы ограничений (2.27), (2.28) равен n . Тогда в соответствии с теоремой 2.9 конус K должен иметь крайние точки. Приведенные выше соображения показывают, что единственной крайней точкой конуса K является его вершина. В этом случае точку X_0 принято называть *острием* конуса K .

Из теоремы 2.7 следует, что K совпадает с множеством точек X вида

$$X = X_0 + \sum_{i=1}^N \beta_i R_i, \quad (2.29)$$

где X_0 — вершина (острие) конуса; R_i — направляющие векторы неограниченных ребер, $\beta_i \geq 0$.

Используя далее теорему 2.10, нетрудно убедиться, что и в общем случае (при любом значении ранга системы условий (2.27), (2.28)) для конуса K имеет место представление (2.29), в котором X_0 — вершина K , а R_i — конечный набор n -мерных векторов. Таким образом, произвольный выпуклый многогранный конус может быть задан в виде множества всех неотрицательных линейных комбинаций конечной системы векторов, сдвинутого на некоторый фиксированный вектор.

Заметим, что понятия вершины многогранного конуса и вершины многогранного множества, вообще говоря, не эквивалентны. Они совпадают только в том случае, если вершина конуса является его острием. В противном случае в качестве вершины конуса может быть принята любая точка, принадлежащая его грани минимальной размерности. Ни одна из этих точек, естественно, не является вершиной конуса, рассматриваемого как многогранное множество.

Отмеченное несоответствие понятий связано с тем, что сложившаяся к настоящему времени терминология для конусов и многогранных множеств создавалась независимо.

Впрочем, это несоответствие не сможет привести к недоразумениям.

Говоря о вершине конуса, мы всегда будем иметь в виду произвольную точку, принадлежащую его грани минимальной размерности.

Выше было дано определение выпуклого многогранного конуса, связанное с первым определением многогранного множества. Если воспользоваться вторым определением многогранного множества, то выпуклому многогранному конусу можно дать другое определение.

Выпуклым многогранным конусом с вершинной в точке X_0 называется совокупность всех неотрицательных линейных комбинаций конечного числа векторов, сдвинутая в точку X_0 .

§ 3. ОСНОВНЫЕ СВОЙСТВА ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

3.1. В последующих двух параграфах устанавливаются основные качественные свойства задачи линейного программирования. При этом существенно используются элементы теории выпуклых многогранных множеств, изложенные в § 2.

Рассмотрим общую задачу линейного программирования, заданную в произвольной форме записи.

Требуется обратить в максимум линейную форму

$$(C, X) = \sum_{j=1}^n c_j x_j \quad (3.1)$$

при условиях

$$(D_i, X) = \sum_{j=1}^n d_{ij} x_j \leq d_i, \quad i = 1, 2, \dots, s, \quad (3.2)$$

$$(D_i, X) = \sum_{j=1}^n d_{ij} x_j = d_i, \quad i = s+1, \dots, s+t. \quad (3.3)$$

Будем предполагать, что система условий (3.2), (3.3) непротиворечива. В таком случае совокупность точек X , удовлетворяющих соотношениям (3.2), (3.3), является выпуклым многогранным множеством. Обозначим его через M . Таким образом, задача (3.1) — (3.3) состоит в максимизации линейной функции (3.1), заданной на многогранном множестве M . Это множество иногда называют *областью определения (задания)* линейной формы задачи или *многогранным множеством условий* данной задачи.

Прежде всего заметим, что ранг системы ограничений (3.2), (3.3) (ранг системы векторов D_i , $i=1, 2, \dots, s+t$) можно всегда считать равным размерности пространства точек X , т. е. равным n . Для того чтобы убедиться в этом, допустим, что ранг системы ограничений (3.2), (3.3) равен $r < n$. Тогда, используя лемму 2.1, приходим к следующему представлению:

$$M = M_0 + \Pi_{n-r}, \quad (3.4)$$

где M_0 — многогранное множество, система ограничений которого имеет ранг, равный n ; Π_{n-r} — $(n-r)$ -мерное подпространство. Возможны два случая:

а) существует точка $Y \in \Pi_{n-r}$, для которой

$$(C, Y) = \sum_{i=1}^n c_i y_i = \zeta \neq 0;$$

б) для любой точки $Y \in \Pi_{n-r}$

$$(C, Y) = 0.$$

Нетрудно заметить, что в случае а) линейная форма (3.1) не ограничена сверху на множестве M (на множестве планов задачи). Действительно, поскольку Π_{n-r} — подпространство, то $\lambda Y \in \Pi_{n-r}$ при любом λ .

Пусть $X_\lambda = X + \lambda Y$, где X — произвольная точка M_0 . В соответствии с представлением (3.4) $X_\lambda \in M$ при любом λ . Рассмотрим

$$(C, X_\lambda) = (C, X) + \lambda (C, Y) = (C, X) + \lambda \zeta.$$

Очевидно, при $\zeta \neq 0$

$$\sup_{\lambda} (C, X_\lambda) = \infty,$$

т. е. линейная форма задачи не ограничена в области своего определения.

Пусть теперь имеет место случай б). Рассмотрим задачу максимизации линейной формы (3.1) при условиях, определяющих многогранное множество M_0 . Если эта задача неразрешима, то, согласно (3.4), задача (3.1) — (3.3) также не имеет решения. Обозначим через M^* и M_0^* совокупности оптимальных планов задач линейного программирования с линейной формой (3.1) и многогранными множествами условий M и M_0 соответственно. Из формулы (3.4), учитывая условие б), получаем, что

$$M^* = M_0^* + P_{n-r}.$$

Таким образом, при решении задачи (3.1) — (3.3) многогранное множество M можно заменить на M_0 . Если вновь полученная задача окажется разрешимой и будет выполнено условие б), то, решив эту задачу, мы получим оптимальный план исходной задачи. В противном случае исходная задача не имеет решения.

Переход от M к M_0 сводится к выделению максимального числа линейно независимых векторов системы D_i , $i = 1, 2, \dots, s+t$. Построение подпространства P_{n-r} осуществляется с помощью разложения остальных векторов D_i по выделенной линейно независимой подсистеме.

В этом параграфе мы будем предполагать, что отмеченные преобразования уже проведены (если, конечно, в них была необходимость) и ранг системы ограничений многогранного множества условий рассматриваемой задачи равен n . Вопрос о необходимости этих преобразований решается обычно на этапе предварительного анализа задачи (в процессе отыскания первого приближения). На том же этапе проводятся и сами преобразования.

Во многих случаях ранг системы (3.2), (3.3) заведомо равен n , так что рассмотренные преобразования оказываются излишними. Это относится, например, к задачам, все переменные которых предполагаются неотрицательными, так как в числе векторов D_i таких задач имеются векторы e_λ , $\lambda = 1, 2, \dots, n$, где

$$e_\lambda = (0, 0, \dots, 0, \underbrace{1}_{\lambda}, 0, \dots, 0).$$

В частности, отсюда следует, что вместо проведения указанных выше преобразований можно просто привести задачу к эквивалентной форме, в которой все переменные предполагаются неотрицательными (например, к канонической форме); ранг системы условий эквивалентной задачи окажется равным n .

3.2. При анализе задач линейного программирования планы, соответствующие вершинам многогранного множества условий, играют особую роль. В этом читатель мог убедиться на примере простейших задач линейного программирования, геометрическая сущность которых была изложена в гл. 1. Поэтому совокупность таких планов естественно выделить.

План $X = (x_1, x_2, \dots, x_n)$ задачи (3.1) — (3.3) будем называть *опорным*, если среди соотношений (3.2), (3.3), которым он удовлетворяет как точным равенствам, имеется n линейно независимых.

Из приведенного определения следует, что понятие опорного плана эквивалентно понятию вершины многогранного множества, определяемого условиями (3.2), (3.3) (см. теорему 2.5). Поэтому число опорных планов задачи линейного программирования всегда конечно. В случае, когда область определения линейной формы задачи оказывается ограниченной, т. е. представляет собой выпуклый многогранник, любой план задачи является выпуклой линейной комбинацией ее опорных планов.

Таким образом, в данном случае выделенная система планов полностью определяет всю совокупность планов задачи, т. е. представляет собой систему опорных точек множества планов, достаточную для восстановления всего этого множества. В связи с этим планы, отвечающие вершинам многогранного множества условий задачи, и были названы опорными.

Следует иметь в виду, что понятие опорного плана не имеет ничего общего с понятием опорной гиперплоскости (см. п. 1.7), которое часто используется при анализе выпуклых множеств.

Поскольку ранг системы ограничений (3.2), (3.3) равен n , то в соответствии с теоремой 2.8 приходим к следующему результату.

Теорема 3.1 (теорема о существовании опорного плана). *Если множество планов задачи (3.1) — (3.3) не пусто, то среди них имеется хотя бы один опорный план.*

Напомним, что решением задачи линейного программирования называется такой ее план, на котором линейная форма задачи достигает условного максимума или минимума (в зависимости от постановки задачи). Задача, обладающая хотя бы одним решением, называется *разрешимой*.

Пусть X_1, X_2, \dots, X_{N_1} — полная совокупность опорных планов задачи (3.1) — (3.3); R_1, R_2, \dots, R_{N_2} — направляющие векторы всех неограниченных ребер многогранного множества M . В таком случае, согласно теореме 2.7, многогранное множество условий задачи (3.1) — (3.3) совпадает с совокупностью точек X вида

$$X = \sum_{i=1}^{N_1} \alpha_i X_i + \sum_{i=1}^{N_2} \beta_i R_i, \quad (3.5)$$

где $\alpha_i \geq 0, \beta_i \geq 0, \sum_{i=1}^{N_1} \alpha_i = 1$.

Предположим, что задача (3.1) — (3.3) разрешима. Тогда, как нетрудно проверить, при любом $i, 1 \leq i \leq N_2$,

$$(C, R_i) \leq 0. \quad (3.6)$$

Действительно, если при некотором $i = i'$

$$(C, R_{i'}) = \delta > 0,$$

то, положив $X(\beta) = X_1 + \beta R_{i'}$, получим

$$\lim_{\beta \rightarrow \infty} (C, X(\beta)) = \infty. \quad (3.7)$$

Но при любом $\beta \geq 0$ вектор $X(\beta)$ — план задачи (3.1) — (3.3). Следовательно, равенство (3.7) означает неразрешимость этой задачи (неограниченность сверху линейной формы задачи на множестве ее планов). Итак, для разрешимых задач неравенство (3.6) выполняется для любого вектора R_i .

Допустим теперь, что X^* — некоторое решение задачи (3.1) — (3.3). Учитывая представление (3.5), справедливое для любого плана задачи, имеем

$$\left. \begin{aligned} X^* &= \sum_{i=1}^{N_1} \alpha_i^* X_i + \sum_{i=1}^{N_2} \beta_i^* R_i, \\ \alpha_i^* &\geq 0, \beta_i^* \geq 0, \sum_{i=1}^{N_1} \alpha_i^* = 1. \end{aligned} \right\} \quad (3.8)$$

Выберем любой индекс $i = i'$, для которого $\alpha_{i'}^* > 0$. Такие индексы заведомо найдутся, поскольку $\sum_{i=1}^{N_1} \alpha_i^* = 1$. Выразим $X_{i'}$ из (3.8):

$$X_{i'} = \frac{1}{\alpha_{i'}^*} \left[X^* - \sum_{i \neq i'} \alpha_i^* X_i - \sum_{i=1}^{N_2} \beta_i^* R_i \right].$$

Умножая обе части последнего равенства скалярно на вектор C , получаем

$$\begin{aligned} (C, X_{i'}) &= \frac{1}{\alpha_{i'}^*} \left[(C, X^*) - \sum_{i \neq i'} \alpha_i^* (C, X_i) - \sum_{i=1}^{N_2} \beta_i^* (C, R_i) \right] \stackrel{(1)}{\geq} \\ &\geq \frac{1}{\alpha_{i'}^*} \left[(C, X^*) - (C, X^*) \sum_{i \neq i'} \alpha_i^* - \sum_{i=1}^{N_2} \beta_i^* (C, R_i) \right] \stackrel{(2)}{\geq} \\ &\geq \frac{1}{\alpha_{i'}^*} \left[(C, X^*) - (C, X^*) \sum_{i \neq i'} \alpha_i^* \right] = \\ &= (C, X^*) \frac{1}{\alpha_{i'}^*} \left(1 - \sum_{i \neq i'} \alpha_i^* \right) \stackrel{(3)}{=} (C, X^*). \end{aligned}$$

Неравенство (1) следует из того, что X^* — решение задачи; неравенство (2) вытекает из (3.6); равенство (3) — следствие условия $\sum_{i=1}^{N_1} \alpha_i^* = 1$.

Итак,

$$(C, X_{i'}) \geq (C, X^*).$$

С другой стороны, так как X^* — решение задачи, то

$$(C, X_{i'}) \leq (C, X^*).$$

Сравнивая два последних неравенства, получаем

$$(C, X_i) = (C, X^*), \quad (3.9)$$

где по условию X_i — некоторый опорный план задачи (4.1) — (4.3).

Назовем решение задачи линейного программирования *опорным*, если оно является опорным планом данной задачи. Соотношение (3.9) эквивалентно следующему утверждению, весьма важному для линейного программирования.

Теорема 3.2 (теорема о существовании опорного решения). *Всякая разрешимая задача линейного программирования, система условий которой имеет ранг n , обладает хотя бы одним опорным решением.*

3.3. Теорема 3.2 подсказывает следующий путь решения задач линейного программирования. Вычислим все опорные планы задачи. Это можно сделать путем исследования C_s^{n-t} систем линейных уравнений, каждая из которых содержит t уравнений (3.3) и какие-то $n-t$ уравнений, отвечающих условиям (3.2). Затем подсчитаем значение линейной формы (3.1) на каждом из полученных опорных планов, число которых конечно. В силу теоремы 3.2 опорный план, соответствующий наибольшему из этих значений, является решением рассматриваемой задачи (если, конечно, она разрешима). Однако при сколь угодно значительных величинах $n-t$ и $s \gg n-t$ (практические задачи обычно удовлетворяют этим условиям) намеченный путь следует признать нереализуемым.

Пусть, например, $t=0$, $s=2n$. Пользуясь формулой Стирлинга, нетрудно подсчитать, что

$$C_{2n}^n \approx \frac{1}{\sqrt{\pi n}} 2^{2n}.$$

Исследование системы n линейных уравнений с n неизвестными (отыскание ее решения или установление линейной зависимости соответствующей системы векторов) занимает порядка n^3 операций. Следовательно, определение всех опорных планов задачи в данном случае

потребуется порядка $\frac{n^3}{\sqrt{n}} 2^{2n}$ операций. Если допустить, что решение задачи осуществляется машиной с быстродействием 10^5 операций в секунду, то при $n=25$ для определения оптимального плана понадобится более 10^6 лет.

Естественно, что способ, приводящий к такому астрономическому объему вычислений даже в сравнительно небольших задачах ($n=25$), не имеет никакой практической ценности. Однако несмотря на несостоятельность метода, основанного на переборе всех вершин многогранного множества условий задачи, сама идея просмотра вершин оказывается весьма полезной. Все конечные методы линейного программирования в той или иной мере связаны с перебором вершин некоторого многогранного множества (не обязательно многогранного множества условий исследуемой задачи). Но этот перебор осуществляется таким образом, что для решения задачи оказывается необходимым просмотреть лишь очень небольшую часть всех имеющихся вершин. Существенное уменьшение сравниваемых вариантов достигается за счет следующих двух свойств, которыми должен обладать любой практически приемлемый процесс перебора:

а) упорядоченность перебора, т. е. невозможность перехода от «лучшей» вершины к «худшей» (понятие «хуже» и «лучше» связано с сущностью метода);

б) наличие критерия, позволяющего без просмотра всех вершин обнаружить, что получена самая «лучшая» вершина (вершина, связанная с решением задачи).

Каждому конечному методу линейного программирования соответствует свой метод упорядочивания перебора и свой критерий окончания перебора.

3.4. Если задача линейного программирования имеет единственное решение, то оно в силу теоремы 3.2 является опорным. Допустим, что исследуемая задача имеет более одного решения. Рассмотрим одно из этих решений, скажем X^* . Для вектора X^* имеет место разложение (3.8). При выводе формулы (3.9) было показано, что если $\alpha_i^* > 0$, то X_i — решение задачи (3.1) — (3.3). Поступая аналогичным образом, нетрудно убедиться в равенстве $(C, R_i) = 0$, справедливом при $\beta_i^* > 0$. Таким образом,

любое решение X^* задачи (3.1) — (3.3) представимо в виде

$$X^* = \sum_{\lambda=1}^{n_1^*} \alpha_{i_\lambda}^* X_{i_\lambda} + \sum_{\lambda=1}^{n_2^*} \beta_{i_\lambda}^* R_{i_\lambda}, \quad (3.10)$$

где $\alpha_{i_\lambda}^* > 0$, $\lambda = 1, 2, \dots, n_1^*$; $\beta_{i_\lambda}^* > 0$, $\lambda = 1, 2, \dots, n_2^*$;

$\sum_{\lambda=1}^{n_1^*} \alpha_{i_\lambda}^* = 1$. При этом

$$(C, X^*) = (C, X_{i_\lambda}), \quad \lambda = 1, 2, \dots, n_1^*, \quad (3.11)$$

$$(C, R_{i_\lambda}) = 0, \quad \lambda = 1, 2, \dots, n_2^*. \quad (3.12)$$

Выделим все опорные решения задачи X_{i_λ} , $\lambda = 1, 2, \dots, n_1$, и все векторы R_{i_λ} , $\lambda = 1, 2, \dots, n_2$, удовлетворяющие условию (3.12). Рассмотрим многогранное множество M^* , порождаемое точками X_{i_λ} и векторами R_{i_λ} , т. е. множество точек X вида

$$X = \sum_{\lambda=1}^{n_1} \alpha_{i_\lambda} X_{i_\lambda} + \sum_{\lambda=1}^{n_2} \beta_{i_\lambda} R_{i_\lambda}, \quad (3.13)$$

где $\alpha_{i_\lambda} \geq 0$, $\lambda = 1, 2, \dots, n_1$; $\beta_{i_\lambda} \geq 0$, $\lambda = 1, 2, \dots, n_2$;
 $\sum_{\lambda=1}^{n_1} \alpha_{i_\lambda} = 1$.

Из соотношений (3.10) — (3.12) вытекает, что любое решение X^* задачи (3.1) — (3.3) содержится в M^* . С другой стороны, поскольку опорные планы X_{i_λ} и векторы R_{i_λ} удовлетворяют условиям (3.11), (3.12) соответственно, то любая точка M^* является решением рассматриваемой задачи. Итак, совокупность всех решений задачи (3.1) — (3.3) является многогранным множеством M^* , состоящим из точек, представимых в виде (3.13). Тем самым мы доказали следующее утверждение.

Теорема 3.3. *Совокупность всех решений задачи линейного программирования является многогранным множеством, порожденным опорными решениями задачи и теми из направляющих векторов R_i неограниченных ребер M , которые удовлетворяют равенству (3.12).*

Из теоремы 3.3, в частности, следует, что задача линейного программирования имеет либо единственное решение, либо бесчисленное множество решений. Однако в последнем случае вся совокупность решений определяется конечным числом векторов X_{i_λ} и R_{i_λ} .

Многогранное множество M^* решений задачи можно задать также, исходя из первого определения выпуклого многогранного множества.

Пусть L^* — оптимальное значение линейной формы (3.1) задачи (3.1) — (3.3). Очевидно, что ограничения (3.2) и (3.3) в совокупности с уравнением

$$\sum_{i=1}^n c_i x_i = L^*$$

составляют систему условий множества M^* .

3.5. Как уже отмечалось в гл. 1, неразрешимость задачи линейного программирования может быть обусловлена либо несовместностью системы условий задачи (множество планов задачи — пустое множество), либо неограниченностью (сверху или снизу в зависимости от постановки задачи) линейной формы задачи на множестве ее планов. Приводимая ниже теорема показывает, что других причин, определяющих неразрешимость задачи линейного программирования, не существует.

Теорема 3.4 (теорема о разрешимости задачи линейного программирования). *Если множество планов задачи линейного программирования не пусто и линейная форма задачи ограничена сверху на этом множестве (речь идет о задаче максимизации), то рассматриваемая задача разрешима, т. е. обладает хотя бы одним решением.*

Доказательство. Поскольку множество планов задачи не пусто, то оно является многогранным множеством и, следовательно, представляет собой совокупность точек X вида

$$X = \sum_{i=1}^{k_1} \alpha_i X_i + \sum_{i=1}^{k_2} \beta_i R_i, \quad (3.14)$$

где все $\alpha_i \geq 0$, $\beta_i \geq 0$; $\sum_{i=1}^{k_1} \alpha_i = 1$; X_i , $i = 1, 2, \dots, k_1$, —

некоторые планы задачи; R_i , $i = 1, 2, \dots, k_2$, — некоторая система векторов.

Отметим, что здесь мы воспользовались общей теоремой о представлении многогранного множества (теорема 2.10), которая не предполагает совпадения ранга системы ограничений, определяющей это множество, с размерностью n пространства точек X . Линейная форма задачи, по предположению, ограничена сверху. Поэтому для любого вектора R_i справедливо неравенство

$$(C, R_i) \leq 0 \quad (3.15)$$

(см. соотношение (3.6)).

Пусть X — произвольный план данной задачи. В силу соотношения (3.14) эта точка может быть представлена в виде суммы

$$X = X' + X'',$$

где

$$X' = \sum_{i=1}^{k_1} \alpha_i X_i, \quad X'' = \sum_{i=1}^{k_2} \beta_i R_i.$$

Учитывая неравенство (3.15) и неотрицательность чисел β_i , получаем

$$(C, X) = (C, X') + (C, X'') \leq (C, X'). \quad (3.16)$$

Выберем точку $X_{i'}$ из условия

$$(C, X_{i'}) = \max_{1 \leq i \leq k_1} (C, X_i).$$

В таком случае, пользуясь свойствами коэффициентов α_i , имеем

$$(C, X') \leq (C, X_{i'}). \quad (3.17)$$

Сравнивая теперь неравенства (3.16) и (3.17), приходим к соотношению

$$(C, X) \leq (C, X_{i'}),$$

справедливому для любого плана X рассматриваемой задачи. Полученное неравенство означает, что план $X_{i'}$ является решением данной задачи. Теорема доказана.

Интересно заметить, что утверждение теоремы 3.4, вообще говоря, справедливо лишь для задач линейного

программирования. Если рассмотреть задачу о максимизации линейной функции, заданной на некотором выпуклом замкнутом множестве D , не являющемся многогранным множеством, то теорема 3.4, вообще говоря, перестает быть верной.

Пусть, например, требуется разыскать максимум формы

$$x_2 \quad (3.18)$$

при условиях

$$x_1 x_2 \leq -1, \quad x_1 \geq 0. \quad (3.19)$$

Ограничения (3.19) высекают в плоскости точек (x_1, x_2) выпуклую замкнутую область D , ограниченную ветвью гиперболы: $x_1 x_2 = -1$, расположенной в четвертом квадранте.

Как нетрудно видеть (рис. 3.9),

$$\sup_{(x_1, x_2) \in D} x_2 = 0,$$

так как $x_2 \leq 0$ и точка $(1/\varepsilon, -\varepsilon) \in D$ при любом $\varepsilon > 0$.

С другой стороны, не существует точки $(x_1, 0) \in D$. Поэтому верхняя грань функции (3.18), определенной на D , не достигается ни в одной точке D .

Для рассматриваемой задачи выполнены все условия теоремы 3.4. Тем не менее эта задача не имеет ни одного решения. Можно привести примеры некоторых непрерывных нелинейных функций, заданных в выпуклых многогранных областях, для которых теорема 3.4 также не имеет места.

Таким образом, предположения о линейности оптимизируемой функции и многогранности ее области определения, фигурирующие в теореме 3.4, являются существенными.

3.6. До сих пор мы рассматривали задачу линейного программирования, заданную в произвольной форме записи (задачу со смешанными условиями). При описании методов линейного программирования чаще всего имеют дело с канонической формой задачи (см. § 3 гл. 1).

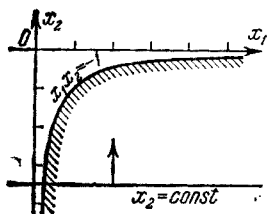


Рис. 3.9.

Обратимся к задаче линейного программирования, заданной в канонической форме.

Требуется максимизировать линейную форму

$$\sum_{j=1}^n c_j x_j \quad (3.20)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (3.21)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n, \quad (3.22)$$

$$m < n.$$

Напомним, что векторы $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$, $j = 1, 2, \dots, n$, принято называть векторами условий, а вектор $B = (b_1, b_2, \dots, b_m)$ — вектором ограничений. Систему равенств (3.21) иногда удобнее записывать в векторной форме:

$$\sum_{j=1}^n A_j x_j = B. \quad (3.21')$$

Задача (3.20) — (3.22) является частным случаем задачи (3.1) — (3.3) при $t = m$, $s = n$:

$$D_i =$$

$$= \begin{cases} (0, 0, \dots, 0, -1, 0, \dots, 0), & i = 1, 2, \dots, n, \\ \underbrace{\hspace{1.5cm}}_i & \\ (a_{i-n,1}, a_{i-n,2}, \dots, a_{i-n,n}), & i = n+1, n+2, \dots, n+m, \end{cases}$$

$$d_i = \begin{cases} 0, & i = 1, 2, \dots, n, \\ b_{i-n}, & i = n+1, n+2, \dots, n+m. \end{cases}$$

Как уже отмечалось, ранг системы ограничений (3.21), (3.22) всегда равен n . Будем предполагать, что все уравнения системы (3.21) линейно независимы. Это предположение не является ограничительным, так как в противном случае часть уравнений системы (3.21) можно было бы отбросить, не изменив многогранное множество условий задачи.

Посмотрим, какую форму примет понятие опорного плана для задачи (3.20) — (3.22).

Пусть $X = (x_1, x_2, \dots, x_n)$ — опорный план этой задачи. В соответствии с определением это означает, что среди векторов D_i , для которых

$$(D_i, X) = d_i, \quad (3.23)$$

имеется n линейно независимых. Векторы D_{i+n} , $i = 1, 2, \dots, m$, линейно независимы и удовлетворяют условию (3.23). Поэтому в случае, если план X — опорный (и только в этом случае), среди векторов D_i , $1 \leq i \leq n$, найдутся такие $n - m$ векторов D_{k_λ} , $1 \leq \lambda \leq n - m$, которые удовлетворяют равенству (3.23) и составляют вместе с векторами D_{i+n} , $1 \leq i \leq m$, линейно независимую систему.

Это значит, что определитель матрицы, строками которой являются векторы $D_{k_\lambda} = -e_{k_\lambda}^T$, $1 \leq \lambda \leq n - m$, и векторы $D_{i+n} = (a_{i1}, a_{i2}, \dots, a_{in})$, $1 \leq i \leq m$, отличен от нуля. Раскладывая этот определитель по элементам первых $n - m$ строк, приходим к условию

$$|(A_{i_1}, A_{i_2}, \dots, A_{i_m})| \neq 0. \quad (3.24)$$

Здесь через $(A_{i_1}, A_{i_2}, \dots, A_{i_m})$ обозначена матрица, составленная из векторов условий $A_{i_1}, A_{i_2}, \dots, A_{i_m}$. Система индексов $I_X = \{i_1, i_2, \dots, i_m\}$ образуется из полной системы $1, 2, \dots, n$ вычеркиванием индексов k_λ для $\lambda = 1, 2, 3, \dots, n - m$. Поскольку $x_{k_\lambda} = 0$ при $1 \leq \lambda \leq n - m$, то при $x_j > 0$ индекс $j \in I_X$. Следовательно, условие (3.24) может быть сформулировано таким образом. Существует линейно независимая система из m векторов условий $A_{i_1}, A_{i_2}, \dots, A_{i_m}$, содержащая все те A_j , для которых $x_j > 0$. Поскольку ранг матрицы (A_1, A_2, \dots, A_n) условий задачи (3.20) — (3.22) равен m , то это требование эквивалентно предположению линейной независимости системы векторов A_j , отвечающих положительным компонентам плана X . Итак, определение опорного плана для задачи линейного программирования, записанной в канонической форме, может быть сформулировано следующим образом.

План $X = (x_1, x_2, \dots, x_n)$ задачи (3.20) — (3.22) называется *опорным*, если векторы условий, отвечающие его положительным составляющим, линейно независимы.

Систему m линейно независимых векторов условий, включающую все те A_j , для которых $x_j > 0$, принято называть *базисом* опорного плана $X = (x_1, x_2, \dots, x_n)$. Составляющие опорного плана, связанные с векторами базиса, иногда называют *базисными компонентами* этого плана.

3.7. При описании конечных методов линейного программирования оказывается полезным выделить некоторый класс задач, состоящий из так называемых невырожденных задач линейного программирования. Приведем соответствующие определения.

Опорный план задачи линейного программирования (3.1) — (3.3) называется *невырожденным*, если число соотношений системы (3.2), (3.3), которым он удовлетворяет как равенствам, равно n . Естественно, что все эти соотношения должны быть линейно независимыми.

Если опорный план не удовлетворяет условию невырожденности, т. е. обращает в равенство более чем n соотношений системы (3.2), (3.3), то его называют *вырожденным планом*.

Задача линейного программирования называется *невырожденной*, если все ее опорные планы являются невырожденными планами. Задачу, имеющую хотя бы один вырожденный опорный план, будем называть *вырожденной задачей*.

Приведенные определения, конечно, имеют смысл лишь при условии, что у рассматриваемой задачи есть опорные планы, т. е. если ранг системы ограничений (3.2), (3.3) равен n .

Если опорный план X задачи (3.1) — (3.3) — невырожденный, то можно легко указать число ребер, выходящих из вершины X многогранного множества условий. Допустим, что

$$(D_i, X) = d_i$$

для

$$i = i_1, i_2, \dots, i_{n-t}, s+1, s+2, \dots, s+t, 1 \leq i_k \leq s.$$

Обозначим множество этих индексов через E_X . По условию

$$(D_i, X) < d_i, \text{ если } i \notin E_X \quad (3.25)$$

(здесь X — невырожденный план). Обозначим через $e^{(\alpha)} = (e_{\alpha 1}, \dots, e_{\alpha n})$, $1 \leq \alpha \leq n-t$, вектор, компоненты которого являются решением системы уравнений

$$(D_i, e^{(\alpha)}) = \begin{cases} 0, & i \in E_X, \quad i \neq i_\alpha, \\ -1, & i = i_\alpha. \end{cases} \quad (3.26)$$

Рассмотрим множество Γ_α точек многогранного множества M , представимых в виде

$$X + \lambda e^{(\alpha)}, \quad \lambda \geq 0.$$

Из уравнений (3.26) и неравенств (3.25) вытекает, что Γ_α содержит некоторый отрезок, т. е. является одномерным множеством. Учитывая далее первые $n-1$ уравнений системы (3.26), приходим к выводу, что Γ_α — ребро множества M с направляющим вектором $e^{(\alpha)}$.

С другой стороны, направляющий вектор e произвольного ребра Γ , выходящего из точки X , обязан удовлетворять системе уравнений

$$(D_i, e) = 0, \quad i \in E_X, \quad i \neq i_\alpha,$$

при некотором α , $1 \leq \alpha \leq n-t$. Поэтому вектор e должен быть параллелен одному из векторов $e^{(\alpha)}$, который, следовательно, можно принять в качестве направляющего для ребра Γ . Итак, ребро Γ совпадает с одним из ребер Γ_α .

Таким образом, из каждой вершины X , являющейся невырожденным планом задачи (3.1)–(3.3), выходит ровно $n-t$ ребер многогранного множества M . Направляющий вектор любого такого ребра может быть определен из системы уравнений (3.26) при некотором значении α , $1 \leq \alpha \leq n-t$. Если опорный план X является вырожденным, то число ребер множества M , выходящих из точки X , не обязано равняться $n-t$. Это число может оказаться как меньшим, так и большим $n-t$.

Геометрический смысл невырожденности опорного плана X состоит в том, что через вершину X проходит ровно n граничных гиперплоскостей многогранного множества M .

Рассмотрим теперь задачу линейного программирования (3.20)–(3.22), записанную в канонической форме. Пусть X — произвольный опорный план этой задачи. Если

векторы условий A_j , $j \in I_X$, составляют базис плана X , то $x_j = 0$ при $j \notin I_X$. Это значит, что план X удовлетворяет, как равенствам, n линейно независимым условиям из системы ограничений, определяющих многогранное множество M :

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

$$x_j = 0, \quad j \notin I_X = \{i_1, i_2, \dots, i_m\}.$$

Условие невырожденности плана X состоит в том, чтобы остальным ограничениям задачи он удовлетворял как строгим неравенствам. Следовательно, для невырожденности плана X необходимо и достаточно, чтобы

$$x_j > 0 \quad \text{при} \quad j \in I_X.$$

Таким образом, определение невырожденного опорного плана задачи линейного программирования, записанной в канонической форме, можно сформулировать следующим образом.

Опорный план X задачи (3.20) — (3.22) называется *невырожденным*, если все его компоненты, отвечающие векторам базиса (базисные компоненты), положительны ($x_j > 0$ при $j \in I_X$). Заметим, что базис невырожденного опорного плана определяется однозначно как система векторов, отвечающих положительным составляющим плана. Вырожденный опорный план может иметь несколько базисов.

§ 4. ГЕОМЕТРИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

4.1. Как мы уже видели в гл. 1, геометрические соображения оказываются весьма полезными при анализе задач линейного программирования. В дальнейшем читатель будет иметь возможность убедиться в том, что геометрические аналоги делают более прозрачными методы линейного программирования и тем самым способствуют их усвоению. Эти аналогии являются также основой эвристических доказательств в линейном программировании. Часто новые подходы к решению задач линейного программирования возникают из элементарных геометрических рассуждений. Во многих случаях геометрические

соображения подсказывают пути строгого обоснования высказанных догадок. В этом параграфе мы опишем две геометрические интерпретации общей задачи линейного программирования, имеющей произвольное число переменных и ограничений. При чтении настоящего параграфа читателю будет полезно обратиться к § 4 гл. 1, где обе эти интерпретации рассматривались применительно к двумерному и трехмерному случаям. Там же были приведены соответствующие рисунки, которые могут послужить хорошей иллюстрацией последующему изложению.

4.2. Начнем с описания первой геометрической интерпретации задачи линейного программирования. Рассмотрим общую задачу линейного программирования (3.1) — (3.3) со смешанными условиями. Система условий задачи (3.2), (3.3) отсекает в n -мерном пространстве точек $X = (x_1, x_2, \dots, x_n)$ выпуклое многогранное множество M (в предположении совместности этой системы). Многогранное множество M может рассматриваться как пересечение (общая часть) полупространств $(D_i, X) \leq d_i$, $i = 1, 2, \dots, s$, и гиперплоскостей $(D_i, X) = d_i$, $i = s+1, s+2, \dots, s+t$.

Граница M складывается из частей граничных гиперплоскостей $(D_i, X) = d_i$, $i = 1, 2, \dots, s+t$. Заметим, что в образовании границы M , вообще говоря, участвуют не все граничные гиперплоскости. Некоторые из них могут не иметь общих точек с M . Естественно, это относится лишь к гиперплоскостям, связанным с условиями системы неравенств (3.2). Ограничения, определяющие подобные гиперплоскости, можно было бы отбросить, не изменив при этом многогранное множество M . Однако аналитический поиск таких ограничений весьма затруднителен.

Размерность p многогранного множества M не превосходит $n-t$, где t — число условий-равенств (3.3), которые предполагаются линейно независимыми. Если все условия (3.2) — нежесткие ограничения M , то $p = n-t$. Поэтому, перенеся начало координат в некоторую точку общей части гиперплоскостей:

$$(D_i, X) = d_i, \quad i = s+1, \dots, s+t,$$

можно рассматривать M в $(n-t)$ -мерном подпространстве основного пространства. Аналитически это может быть осуществлено путем выражения из уравнений (3.3) каких-то t переменных (через остальные $n-t$) с последующим исключением их из неравенств (3.2). Таким образом, размерность пространства, содержащего M , понижается до $n-t$. Этим приемом мы уже пользовались в § 4 гл. 1. Пусть вектор $C = (c_1, c_2, \dots, c_n)$, определяющий линейную форму задачи, отличен от нулевого. В таком случае линейная форма (3.1) задачи задает в n -мерном пространстве семейство параллельных гиперплоскостей

$$(C, X) = \lambda, \quad -\infty < \lambda < \infty. \quad (4.1)$$

Каждую из этих гиперплоскостей будем называть *гиперплоскостью линейной формы* задачи. Коэффициенты линейной формы c_j , $j = 1, 2, \dots, n$, составляют вектор $C = (c_1, c_2, \dots, c_n)$, ортогональный семейству гиперплоскостей (4.1). Вектор C указывает направление возрастания линейной формы задачи. При фиксированном значении параметра λ гиперплоскость линейной формы определяет два полупространства. То из них, которое содержит точку $X + C$ (точка X принадлежит гиперплоскости), назовем *верхним полупространством*. Другое полупространство будем называть *нижним полупространством*. Уравнение верхнего полупространства имеет вид

$$(C, X) \geq \lambda.$$

Нижнее полупространство определяется уравнением

$$(C, X) \leq \lambda.$$

Пусть при $\lambda = \lambda_0$ гиперплоскость линейной формы имеет общие точки с многогранным множеством условий M (или M содержится в верхнем полупространстве этой гиперплоскости). Значения линейной формы во всех этих точках совпадают. Передвигая гиперплоскость параллельно самой себе в сторону, определяемую вектором C (в сторону возрастания линейной формы (3.1)), можно прийти к такому ее положению, когда при дальнейшем смещении гиперплоскость уже не будет иметь общих точек с M . Если множество M содержится в нижнем полупространстве исходной гиперплоскости линейной формы,

то движение осуществляется в направлении вектора $-C$ (в сторону убывания линейной формы задачи).

Пусть полученное предельное положение гиперплоскости отвечает значению параметра λ , равному λ_1 , т. е. соответствующая гиперплоскость имеет уравнение

$$(C, X) = \lambda_1. \quad (4.2)$$

В этом случае многогранное множество условий M расположено в нижнем полупространстве гиперплоскости (4.2). Во всех точках, принадлежащих как M , так и гиперплоскости (4.2) (множество таких точек не пусто по построению), линейная форма принимает экстремальное значение λ_1 . Общая часть M и гиперплоскости (4.2) определяет многогранное множество M^* решений данной задачи.

Если задача имеет единственное решение, то M^* состоит из единственной точки — вершины M . В общем случае M^* — некоторое выпуклое многогранное множество, размерность ρ^* которого удовлетворяет неравенствам

$$0 \leq \rho^* \leq \rho.$$

Отметим, что равенство $\rho^* = \rho$ имеет место лишь в том случае когда M содержится в одной из гиперплоскостей семейства (4.1).

До сих пор мы предполагали, что существует предельное положение гиперплоскости линейной формы, определяемое уравнением (4.2). Очевидно, это предположение оправдано при ограниченности M (если M — выпуклый многогранник). Если же M представляет собой неограниченное выпуклое многогранное множество, то может случиться, что при сколь угодно далеком смещении гиперплоскости линейной формы в сторону, определяемую вектором C , она будет иметь общие точки с M . Это означает, что линейная форма задачи не ограничена на множестве M , т. е. данная задача неразрешима.

Следует иметь в виду, что неограниченность M не является достаточным условием неразрешимости задачи. При одних значениях вектора C задача с неограниченным многогранным множеством условий может быть разрешимой (гиперплоскость линейной формы имеет конечное предельное положение), при других — неразрешимой

(конечного предельного положения гиперплоскости линейной формы не существует).

Иногда некорректная постановка задачи приводит к несовместности условий (3.2), (3.3). Геометрически этот случай соответствует тому, что область определения линейной формы задачи вырождается в пустое множество.

4.3. Первая геометрическая интерпретация задачи линейного программирования не чувствительна к форме записи задачи. Вторая геометрическая интерпретация, к описанию которой мы переходим, приспособлена к задачам в канонической форме.

Итак, рассмотрим общую задачу линейного программирования в канонической форме (задача (3.20) — (3.22)).

Введем новые переменные $u_1, u_2, \dots, u_m, u_{m+1}$:

$$\left. \begin{aligned} u_i &= \sum_{j=1}^n a_{ij} x_j, \quad i = 1, 2, \dots, m, \\ u_{m+1} &= \sum_{j=1}^n c_j x_j. \end{aligned} \right\} \quad (4.3)$$

Соотношения (4.3) определяют преобразование n -мерного пространства точек $X = (x_1, x_2, \dots, x_n)$ в $(m+1)$ -мерное пространство точек $U = (u_1, u_2, \dots, u_{m+1})$. Через \bar{A}_j обозначим $(m+1)$ -мерный вектор-столбец с компонентами $a_{1j}, a_{2j}, \dots, a_{mj}, c_j$, т. е. $\bar{A}_j = (a_{1j}, a_{2j}, \dots, a_{mj}, c_j)^T$. Вектор \bar{A}_j , первые m компонент которого совпадают с компонентами вектора условий A_j , был назван в гл. 1 расширенным вектором условий. Теперь соотношения (4.3) можно переписать в векторной форме:

$$U = \sum_{j=1}^n \bar{A}_j x_j. \quad (4.4)$$

По определению множество точек U , представимых в виде (4.4) при $x_j \geq 0, j = 1, 2, \dots, n$, является выпуклым многогранным конусом (см. второе определение многогранного конуса в п. 2.7). Обозначим этот конус через K . Конус K порожден расширенными векторами условий \bar{A}_j ; вершина K расположена в начале координат. Из соотношения (4.4) следует, что K является

образом положительного ортанта *) пространства точек X в $(m+1)$ -мерном пространстве точек U .

Пусть точка X удовлетворяет системе равенств (3.21). Тогда в соответствии с (4.4) образом этой точки в рассматриваемом $(m+1)$ -мерном пространстве является точка $U_X = (b_1, b_2, \dots, b_m, L(X))$, где $L(X) = \sum_{j=1}^n c_j x_j$. С другой стороны, при любом λ среди решений системы (3.21) найдется такое решение X , что

$$\lambda = L(X) = \sum_{j=1}^n c_j x_j.$$

(Последнее утверждение, очевидно, имеет место, если вектор C не является линейной комбинацией векторов $(a_{i1}, a_{i2}, \dots, a_{in})$, $i=1, 2, \dots, m$. Заметим, что случай, когда это условие не выполняется, мало интересен, так как он эквивалентен предположению о постоянстве линейной формы (3.21) на многогранном множестве условий задачи). Следовательно, преобразование (4.4) переводит совокупность решений системы (3.21) в прямую Q , уравнение которой имеет вид

$$U = \bar{B} + \lambda e_{m+1}, \quad -\infty < \lambda < \infty, \quad (4.5)$$

где $\bar{B} = (b_1, b_2, \dots, b_m, 0)$; $e_{m+1} = (0, 0, \dots, 0, 1)$. Прямая Q проходит через точку \bar{B} и параллельна оси Ou_{m+1} . Поскольку планы задачи (3.20)–(3.22) обязаны удовлетворять как равенствам (3.21), так и неравенствам (3.22), преобразование (4.4) переводит многогранное множество условий задачи M в общую часть конуса K и прямой Q .

Допустим, что M не является пустым множеством. Тогда конус K и прямая Q имеют общие точки. Обозначим пересечение прямой Q и конуса K через Q_k . Пересечение выпуклых множеств выпукло. Множество Q_k имеет размерность, меньшую или равную единице. Следовательно, Q_k — прямая, либо полупрямая (луч), либо

*) Положительным ортантом принято называть n -мерный аналог положительного октанта, т. е. совокупность векторов X с неотрицательными координатами,

отрезок, который может вырождаться в точку. Каждой точке $U = (b_1, b_2, \dots, b_m, \lambda) \in Q_k$ соответствует совокупность точек $X \in M$, для которых $(C, X) = \lambda$.

Изучаемая задача линейного программирования состоит в определении такой точки $X^* \in M$, на которой (C, X) достигает своего максимума. В рассматриваемом $(m+1)$ -мерном пространстве решение задачи эквивалентно отысканию точки $U^* \in Q_k$ с максимально возможной $(m+1)$ -й координатой λ^* . Здесь следует различать два случая:

а) существует такое $\bar{\lambda}$, что $\lambda \leq \bar{\lambda}$ для любой точки $(b_1, b_2, \dots, b_m, \lambda) \in Q_k$;

б) множество Q_k содержит точки со сколь угодно большими значениями $(m+1)$ -й координаты.

В первом случае Q_k представляет собой либо отрезок, либо луч с направляющим вектором $-e_{m+1}$ (луч, идущий вдоль отрицательного направления оси Ou_{m+1}). При этом исследуемая задача оказывается разрешимой, и ее оптимальному многогранному множеству M^* (совокупности решений) соответствует верхний (в смысле оси Ou_{m+1}) конец Q_k — верхняя точка пересечения конуса K и прямой Q .

Если множество Q_k является прямой или лучом с направляющим вектором e_{m+1} , то приходим к случаю б). Наличие этого случая, очевидно, означает, что линейная форма задачи не ограничена на множестве планов задачи.

Может оказаться, что прямая Q проходит вне конуса K , и, следовательно, общая часть Q и K — пустое множество. В этом случае задача не имеет ни одного плана, т. е. система условий (3.21), (3.22) противоречива.

Итак, решение задачи (3.20) — (3.22) в терминах второй геометрической интерпретации состоит в отыскании верхней точки пересечения прямой Q и конуса K . При отсутствии общих точек у Q и K задача оказывается неразрешимой из-за несовместности ее условий (3.21), (3.22). Если же общая часть Q и K — непустое множество, не имеющее самой верхней точки, то задача неразрешима вследствие неограниченности линейной формы (3.20) на многогранном множестве условий M .

При описании геометрической сущности задачи линейного программирования мы ограничились задачами

максимизации. Все сказанное выше с точностью до совершенно естественных изменений справедливо и для задач минимизации. При желании читатель может проинвестировать эти изменения самостоятельно.

§ 5. ПРЯМАЯ И ДВОЙСТВЕННАЯ ЗАДАЧИ

5.1. Этот и следующие параграфы настоящей главы посвящены изложению одного из центральных пунктов линейного программирования — теории двойственности.

Произвольной задаче линейного программирования можно определенным образом сопоставить некоторую другую задачу (опять-таки линейного программирования), называемую обычно двойственной или сопряженной. Теория двойственности обнаруживает тесную связь между обеими задачами, составляющими единую двойственную пару. Совместный анализ пары двойственных задач оказывается плодотворным как при построении численных методов линейного программирования, так и при различных качественных исследованиях.

Рассмотрим каноническую форму общей задачи линейного программирования, состоящую, как обычно, в максимизации линейной формы

$$L(X) = \sum_{j=1}^n c_j x_j \quad (5.1)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (5.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (5.3)$$

Одновременно с задачей (5.1) — (5.3) введем в рассмотрение задачу минимизации линейной формы

$$\tilde{L}(Y) = \sum_{i=1}^m b_i y_i \quad (5.4)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n. \quad (5.5)$$

Задачу (5.4), (5.5) принято называть *двойственной* по отношению к задаче (5.1) — (5.3) или *сопряженной* с ней. Задачу (5.1) — (5.3) при этом называют *прямой* задачей.

Перепишем условия сформулированных задач в матричной форме. Обозначим через $A = \|a_{ij}\| = (A_1, \dots, A_n)$ матрицу условий задачи (5.1) — (5.3), составленную из векторов условий A_j этой задачи. Положим, как обычно,

$$B = (b_1, b_2, \dots, b_m)^T, \quad C = (c_1, c_2, \dots, c_n)^T.$$

В этих обозначениях прямая и двойственная задачи принимают следующий вид.

Прямая задача. Требуется определить n -мерный вектор $X = (x_1, x_2, \dots, x_n)^T$, обращающий в максимум

$$L(X) = (C, X) \quad (5.1')$$

при условиях

$$AX = B, \quad (5.2')$$

$$X \geq 0. \quad (5.3')$$

Двойственная задача. Требуется определить m -мерный вектор $Y = (y_1, y_2, \dots, y_m)^T$, сбрашающий в минимум

$$\tilde{L}(Y) = (B, Y) \quad (5.4')$$

при условиях

$$A^T Y \geq C. \quad (5.5')$$

Рассмотрим, например, задачу линейного программирования, состоящую в максимизации линейной формы

$$L(X) = x_1 + 3x_2 + 2x_3 - 3x_5 - x_6 \quad (5.6)$$

при условиях

$$\left. \begin{aligned} 2x_1 + 2x_2 + x_3 + x_4 + 2x_5 + x_6 &= 1, \\ 4x_1 + 3x_2 - x_3 - 2x_4 - x_5 + 2x_6 &= 1, \end{aligned} \right\} \quad (5.7)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, 6. \quad (5.8)$$

Для данной задачи

$$A = \begin{pmatrix} 2 & 2 & 1 & 1 & 2 & 1 \\ 4 & 3 & -1 & -2 & -1 & 2 \end{pmatrix}, \quad B = (1, 1)^T,$$

$$C = (1, 3, 2, 0, -3, -1).$$

В соответствии с общими правилами задача, двойственная по отношению к задаче (5.6) — (5.8), формулируется следующим образом.

Требуется обратить в минимум линейную форму

$$\tilde{L}(Y) = y_1 + y_2 \quad (5.9)$$

при условиях

$$\left. \begin{aligned} 2y_1 + 4y_2 &\geq 1, \\ 2y_1 + 3y_2 &\geq 3, \\ y_1 - y_2 &\geq 2, \\ y_1 - 2y_2 &\geq 0, \\ 2y_1 - y_2 &\geq -3, \\ y_1 + 2y_2 &\geq -1. \end{aligned} \right\} \quad (5.10)$$

5.2. Приведем геометрическое истолкование двойственной задачи. Для этого предварительно напомним геометрический смысл прямой задачи в терминах второй геометрической интерпретации (см. § 4). Условия задачи (5.1) — (5.3) задают линейное преобразование n -мерного пространства точек $X = (x_1, x_2, \dots, x_n)$ в $(m+1)$ -мерное пространство точек $U = (u_1, u_2, \dots, u_{m+1})$:

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

$$u_{m+1} = \sum_{j=1}^n c_j x_j.$$

Образом положительного ортанта пространства точек X является выпуклый многогранный конус K с вершиной в начале координат, порожденный расширенными векторами условий $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n$, где, как обычно,

$$\bar{A}_j = (a_{1j}, a_{2j}, \dots, a_{mj}, c_j)^T.$$

Образом совокупности решений системы (5.2) является прямая Q , проходящая через точку $\bar{B} = (b_1, b_2, \dots, b_m, 0)$ параллельно координатной оси Ou_{m+1} . Уравнение прямой Q имеет вид

$$U = \bar{B} + \lambda e_{m+1}, \quad -\infty < \lambda < \infty, \quad (5.11)$$

где $e_{m+1} = \underbrace{(0, 0, \dots, 0)}_m, 1$.

Задача (5.1) — (5.3) состоит в отыскании «верхней» точки пересечения прямой Q и конуса K .

Рассмотрим совокупность гиперплоскостей $(m+1)$ -мерного пространства точек U , проходящих через начало координат. Уравнение любой такой гиперплоскости имеет вид

$$\sum_{k=1}^{m+1} \lambda_k u_k = 0. \quad (5.12)$$

Гиперплоскость (5.12) и ее направляющий вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m, \lambda_{m+1})$ связаны условием ортогональности.

Направляющий вектор определяется с точностью до произвольного множителя (отличного от нуля). Для наших целей удобно доопределить направляющий вектор Λ условием

$$\lambda_{m+1} = -1. \quad (5.13)$$

Тем самым гиперплоскости, параллельные координатной оси Ou_{m+1} , исключаются из рассмотрения. Если выполнено условие (5.13), то устанавливается взаимно однозначное соответствие между гиперплоскостями, проходящими через начало координат и не содержащими координатную ось Ou_{m+1} , и их направляющими векторами.

Пусть $Y = (y_1, y_2, \dots, y_m)$ — произвольный план задачи (5.4), (5.5). Рассмотрим гиперплоскость P_Y , определяемому уравнением

$$\sum_{i=1}^m y_i u_i - u_{m+1} = 0. \quad (5.14)$$

Проверим, что конус K лежит в одном из полупространств, порожденных P_Y .

Действительно, в соответствии с условиями (5.5), которым удовлетворяет вектор Y , результат подстановки в левую часть (5.14) координат любого вектора \bar{A}_j ($j=1, 2, \dots, n$) является неотрицательным числом. Поэтому конус K , порожденный векторами \bar{A}_j , $j=1, 2, \dots, m$, лежит по ту же сторону от гиперплоскости P_Y , что и вектор $-e_{m+1} = (0, 0, \dots, 0, -1)$, соответствующий отрицательному направлению оси Ou_{m+1} . Таким образом,

конус K расположен «под» гиперплоскостью Π_Y (в смысле координатной оси Ou_{m+1}).

Рассмотрим теперь произвольную гиперплоскость, проходящую через начало координат и не содержащую координатной оси Ou_{m+1} . Пусть ее уравнение имеет вид (5.14). Если конус K расположен «под» гиперплоскостью (5.14), то вектор $Y = (y_1, y_2, \dots, y_m)$, определяющий эту гиперплоскость, удовлетворяет условиям (5.5) и, следовательно, является планом двойственной задачи (5.4), (5.5). Для того чтобы убедиться в этом, достаточно подставить в левую часть (5.14) координаты расширенного вектора условий \bar{A}_j ($j = 1, 2, \dots, n$), который, по предположению, расположен «под» рассматриваемой гиперплоскостью.

Итак, геометрическим образом множества планов двойственной задачи является совокупность гиперплоскостей, содержащих начало координат и расположенных «над» конусом K . При этом между планами Y двойственной задачи и гиперплоскостями Π_Y указанной совокупности устанавливается взаимно однозначное соответствие, определяемое уравнением (5.14).

Найдем значение $(m+1)$ -й координаты $u_{m+1}^{(Y)}$ точки пересечения прямой Q и гиперплоскости Π_Y . Используя уравнения (5.11) и (5.14), имеем

$$u_{m+1}^{(Y)} = \sum_{i=1}^m b_i y_i = \tilde{L}(Y). \quad (5.15)$$

Соотношение (5.15) означает, что значение линейной формы двойственной задачи на плане Y равно «расстоянию» точки пересечения прямой Q и гиперплоскости Π_Y до гиперплоскости $u_{m+1} = 0$ (слово «расстояние» взято в кавычки вследствие того, что $u_{m+1}^{(Y)}$ может быть как положительным, так и отрицательным числом).

Теперь мы можем указать геометрический смысл двойственной задачи. С геометрической точки зрения двойственная задача (5.4), (5.5) заключается в отыскании такой гиперплоскости, содержащей начало координат и расположенной «над» конусом K , которая пересекает прямую Q в «наинижней» точке (в смысле оси Ou_{m+1}).

5.3. При $m=2$ все сказанное в предыдущем пункте приобретает геометрическую наглядность.

Обратимся к задаче (5.6)–(5.8), у которой число условий равенств (5.7) равно 2. В данном случае расширенные векторы условий имеют такой вид:

$$\begin{aligned}\bar{A}_1 &= \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}, & \bar{A}_2 &= \begin{pmatrix} 2 \\ 3 \\ 3 \end{pmatrix}, & \bar{A}_3 &= \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}, \\ \bar{A}_4 &= \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}, & \bar{A}_5 &= \begin{pmatrix} 2 \\ -1 \\ -3 \end{pmatrix}, & \bar{A}_6 &= \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}.\end{aligned}$$

Конус K образован векторами \bar{A}_j , $j=1, 2, 3, 4, 5, 6$, и расположен в трехмерном пространстве точек $U = (u_1, u_2, u_3)$. Ребрами конуса K являются лучи

$$U = \bar{A}_j \lambda, \quad \lambda \geq 0,$$

$$j=1, 2, 3, 4, 5, 6.$$

Для прямой Q , согласно соотношениям (5.11), имеем

$$\left. \begin{aligned} u_1 &= 1, & u_2 &= 1, \\ u_3 &= \lambda, \end{aligned} \right\} \quad (5.16)$$

где $-\infty < \lambda < \infty$.

Пересечем конус K плоскостью, проходящей через точку $(1, 0, 0)$ и перпендикулярной координатной оси Ou_1 . Уравнение этой плоскости будет

$$u_1 = 1.$$

Общая часть плоскости $u_1 = 1$ и конуса K представляет собой многоугольник $a_1 a_2 a_3 a_4 a_5 a_6$. Вершины многоугольника $a_1 a_2 a_3 a_4 a_5 a_6$ являются точками пересечения плоскости $u_1 = 1$ с соответствующими ребрами $U = \bar{A}_j \lambda$, $\lambda \geq 0$, многогранного конуса K . Многоугольник $a_1 a_2 a_3 a_4 a_5 a_6$

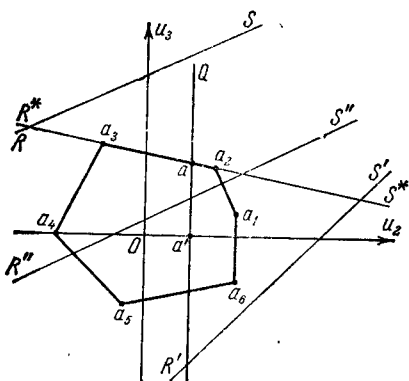


Рис. 3.10.

изображен на рис. 3.10. На этом рисунке изображена также прямая Q , содержащаяся в плоскости $u_1 = 1$.

Каждая плоскость, проходящая через начало координат и не содержащая координатную ось Ou_3 , пересекает плоскость $u_1 = 1$ по прямой, не параллельной оси Q . Будем называть такие прямые следами соответствующих плоскостей. На рис. 3.10 изображены четыре подобных следа:

$$RS, R'S', R''S'', R^*S^*.$$

Как нетрудно заметить, произвольная прямая на рис. 3.10 оказывается следом плоскости, соответствующей плану двойственной задачи (5.9), (5.10), в том и только в том случае, если многоугольник $a_1a_2a_3a_4a_5a_6$ расположен под этой прямой. В частности, прямые RS, R^*S^* соответствуют планам двойственной задачи. Остальные две прямые не определяют планов двойственной задачи: прямая $R'S'$ проходит ниже многоугольника $a_1a_2a_3a_4a_5a_6$, прямая $R''S''$ делит этот многоугольник на две части. Геометрическим образом решения прямой задачи является точка a — верхняя точка пересечения прямой Q и многоугольника $a_1a_2a_3a_4a_5a_6$.

Процесс решения двойственной задачи состоит в выборе такой прямой, которая расположена над многоугольником $a_1a_2a_3a_4a_5a_6$ и пересекает ось Q возможно ниже. Геометрически очевидно, что этим свойством обладает прямая R^*S^* , являющаяся опорной для многоугольника $a_1a_2a_3a_4a_5a_6$ в точке a . Отсюда следует совпадение значений линейных форм задач (5.6) — (5.8) и (5.9), (5.10) на их оптимальных планах. Оптимальная величина обеих линейных форм равна длине отрезка aa' (см. рис. 3.10). В дальнейшем мы убедимся, что отмеченные здесь свойства задач (5.6) — (5.8) и (5.9), (5.10) справедливы для произвольных задач линейного программирования.

При выяснении геометрического смысла задачи (5.9), (5.10) мы, естественно, могли бы оперировать с самим конусом K , а не с сечением этого конуса плоскостью $u_1 = 1$, содержащей ось Q . Переход к многоугольнику $a_1a_2a_3a_4a_5a_6$ был осуществлен здесь лишь для того, чтобы сделать изложение более наглядным. Отметим, что вместо плоскости $u_1 = 1$ можно было бы взять любую плоскость, содержащую прямую Q .

5.4. Рассмотрим теперь произвольную задачу линейного программирования с однотипными условиями.

Требуется обратить в максимум линейную форму

$$\sum_{j=1}^n c_j x_j \quad (5.17)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m, \quad (5.18)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (5.19)$$

Задачу (5.17) — (5.19) нетрудно привести к канонической форме. Для этого достаточно ввести дополнительные неотрицательные переменные x_{n+i} , $i = 1, 2, \dots, m$, и переписать условия задачи (5.17) — (5.19) в эквивалентном виде

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i, \quad i = 1, 2, \dots, m, \quad (5.18')$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n + m. \quad (5.19')$$

Эквивалентная задача имеет $n + m$ неотрицательных переменных и m условий-равенств.

Сформулируем, руководствуясь общими правилами, задачу, двойственную по отношению к задаче (5.17), (5.18'), (5.19').

Расширенные векторы условий прямой задачи имеют вид

$$\bar{A}_j = \begin{cases} (a_{1j}, a_{2j}, \dots, a_{mj}, c_j)^T, & \text{если } 1 \leq j \leq n, \\ (0, 0, \dots, 0, 1, 0, \dots, 0)^T, & \text{если } n+1 \leq j \leq n+m. \end{cases}$$

$\underbrace{\hspace{10em}}_{j-n}$

Поэтому двойственная задача формулируется следующим образом.

Требуется обратить в минимум линейную форму

$$\sum_{i=1}^m b_i y_i \quad (5.20)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n, \quad (5.21)$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m. \quad (5.22)$$

Важно отметить, что двойственная задача (5.20) — (5.22) является задачей с однотипными условиями. Если ввести новые параметры $a'_{ij} = -a_{ij}$, $b'_i = -b_i$, $c'_j = -c_j$, то задача (5.20) — (5.22) превращается в задачу максимизации линейной формы

$$\sum_{i=1}^m b'_i y_i \quad (5.20')$$

при условиях

$$\sum_{i=1}^m a'_{ij} y_i \leq c'_j, \quad j = 1, 2, \dots, n, \quad (5.21')$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m. \quad (5.22')$$

Полученная задача имеет точно такой же вид, как и задача (5.17) — (5.19). Следовательно, двойственной по отношению к ней является задача минимизации линейной формы

$$\sum_{j=1}^n c'_j x_j \quad (5.23)$$

при условиях

$$\sum_{j=1}^n a'_{ij} x_j \geq b'_i, \quad i = 1, 2, \dots, m, \quad (5.24)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (5.25)$$

Если теперь в задаче (5.23) — (5.25) перейти к старым параметрам a_{ij} , b_i , c_j , то получим прямую задачу (5.17) — (5.19).

Таким образом, прямая задача (5.17) — (5.19) является сопряженной к двойственной задаче (5.20) — (5.22). В связи с этим задачи (5.17) — (5.19) и (5.20) — (5.22) естественно называть *двойственной* или *взаимосопряженной парой*. Каждая из задач этой пары двойственна по отношению

к другой задаче. Поэтому выделение прямой задачи из взаимосопряженной пары носит чисто условный характер. В дальнейшем мы убедимся, что задачи (5.1) — (5.3) и (5.4), (5.5) также составляют двойственную пару.

5.5. Приведем экономическое истолкование задачи (5.20) — (5.22), двойственной по отношению к задаче (5.17) — (5.19). Прежде всего напомним экономическую интерпретацию прямой задачи (5.17) — (5.19), которая была изложена в § 5 гл. 1.

Имеется n способов производства некоторого однородного продукта. Количество продукта, вырабатываемое с помощью j -го способа производства в единицу времени, составляет c_j единиц. Использование j -го способа производства в течение единицы времени связано с расходом i -го фактора производства ($i = 1, 2, \dots, m$), равным величине a_{ij} . Допустим, что запасы факторов производства составляют соответственно b_1, b_2, \dots, b_m единиц. Время, в течение которого производство ведется по j -му способу, обозначим через x_j .

Задача (5.17) — (5.19) является математической формулировкой проблемы составления такого плана использования различных способов производства, который позволяет получить максимальное количество однородного продукта при имеющихся в наличии ресурсах (b_1, b_2, \dots, b_m).

Допустим теперь, что необходимо, оставаясь в рамках рассматриваемого производства, оценить каждый из его факторов. Мы здесь будем рассматривать лишь идеализированную замкнутую модель производства, в которой связи с внешним миром строго фиксированы условиями задачи.

Условия задачи (ограниченные ресурсы и отработанные способы производства) определяют оценку каждого фактора, внутреннюю для данного производства. Следует иметь в виду, что эта оценка является относительной. Одни и те же производственные факторы для разных предприятий и районов представляют различную ценность. Изменение условий производства, в частности изменение запасов различных факторов, приводит к необходимости переоценки этих факторов. Относительность оценок факторов производства связана также и с тем, что эти оценки

измеряются в единицах ценности выпускаемой продукции. Ценность продукции определяется условиями, внешними по отношению к данному производству. Примем оценку единицы производимого продукта за единицу. Оценка единицы производимого продукта (ценность продукции) является здесь исходным понятием, отправляясь от которого, можно установить оценки различных факторов производства.

Обозначим через y_i ($i = 1, 2, \dots, m$) оценку единицы i -го фактора производства (оценку i -го фактора).

Исследуем j -й способ производства с точки зрения расходов и доходов. Если использовать этот способ производства в течение единицы времени, то оценка всех затрат составит

$$z_j = \sum_{i=1}^m a_{ij} y_i,$$

а оценка полученной продукции окажется равной c_j . При правильно выбранных оценках факторов производства оценка суммарных расходов не может быть меньше оценки полученной продукции, ибо в противном случае часть продукции была бы создана из «ничего».

Следовательно, для любого $j = 1, 2, \dots, n$

$$z_j \geq c_j.$$

Другими словами, вектор оценок

$$Y = (y_1, y_2, \dots, y_m)$$

должен подчиняться условиям (5.21). Кроме того, оценки y_1, y_2, \dots, y_m естественно считать неотрицательными числами. Поэтому вектор

$$Y = (y_1, y_2, \dots, y_m)$$

удовлетворяет также условиям (5.22).

Итак, вектор оценок факторов производства является планом двойственной задачи (5.20) — (5.22). Однако условия (5.21), (5.22) не могут полностью определить вектор оценок Y . В рассматриваемой экономической интерпретации задачи (5.17) — (5.19) параметры a_{ij} естественно считать неотрицательными числами, причем при любом j хотя бы одна из величин a_{ij} должна быть отлична от

нуля. Поэтому любой вектор с достаточно большими компонентами является планом двойственной задачи (5.20) — (5.22). Возникает необходимость в условии, не допускающем необоснованного завышения оценок факторов производства. Естественным ограничением подобного типа является следующее. Вектор оценок Y должен быть таким, чтобы суммарная оценка ресурсов

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

которыми располагает анализируемое производство, была возможно меньшей.

В дальнейшем мы увидим, что если это условие не выполняется, то при любом плане использования способов производства оценка полученной продукции меньше суммарной оценки всех ресурсов. Напротив, при учете указанного условия существуют планы производства, обеспечивающие равенство оценок произведенной продукции и имеющихся ресурсов. Это обстоятельство оправдывает введение последнего ограничения.

Таким образом, двойственная задача (5.20) — (5.22) является математической формулировкой проблемы правильной оценки всех факторов производства. Вектор оценок факторов производства совпадает с решением двойственной задачи. В дальнейшем мы будем иногда называть планы двойственной задачи векторами *предварительных оценок* факторов производства.

Пару двойственных задач (5.17) — (5.19) и (5.20) — (5.22) удобно задавать в виде наглядной таблицы (табл. 3.1).

5.6. В заключение параграфа приведем несколько простых, но весьма полезных утверждений относительно планов прямой и двойственной задач. Эти утверждения будут неоднократно использоваться в последующем изложении.

Лемма 5.1. Если $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$ — произвольные планы задач (5.1) — (5.3) и (5.4), (5.5) соответственно, то

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i. \quad (5.26)$$

Таблица 3.1

		Производительность							
		$c_1 \quad c_2 \quad . . . \quad c_j \quad . . . \quad c_n$							
Вектор оценок	y_1	a_{11}	a_{12}	$. . .$	a_{1j}	$. . .$	a_{1n}	b_1	Ресурсы
	y_2	a_{21}	a_{22}	$. . .$	a_{2j}	$. . .$	a_{2n}	b_2	
	\dots	$. . .$	$. . .$	$. . .$	$. . .$	$. . .$	$. . .$	\dots	
	y_i	a_{i1}	a_{i2}	$. . .$	a_{ij}	$. . .$	a_{in}	b_i	
	\dots	$. . .$	$. . .$	$. . .$	$. . .$	$. . .$	$. . .$	\dots	
	y_m	a_{m1}	a_{m2}	$. . .$	a_{mj}	$. . .$	a_{mn}	b_m	
		$x_1 \quad x_2 \quad . . . \quad x_j \quad . . . \quad x_n$							
		План производства							

Доказательство. По условию

$$\sum_{i=1}^m a_{ij} y_i \geq c_j \quad \text{для } j = 1, 2, \dots, n.$$

Следовательно,

$$\sum_{j=1}^n c_j x_j \leq \sum_{j=1}^n x_j \sum_{i=1}^m a_{ij} y_i = \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j.$$

Вектор X является планом задачи (5.1) — (5.3). Поэтому

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

Преобразуя с помощью этих равенств правую часть предыдущего соотношения, имеем

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i.$$

Лемма доказана. Из леммы 5.1, в частности, следует, что, приняв в качестве вектора оценок план двойственной задачи, не являющийся ее решением, мы не сможем получить совпадения оценок произведенной продукции и имеющихся ресурсов.

Действительно, если Y — неоптимальный план двойственной задачи, а Y^* — решение этой задачи, то при любом плане X прямой задачи

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i^* < \sum_{i=1}^m b_i y_i.$$

Лемма 5.2. Если для некоторых планов $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ и $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$ задач (5.1) — (5.3) и (5.4), (5.5) соответственно выполняется равенство

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*, \quad (5.27)$$

то векторы X^* , Y^* являются решениями соответствующих задач.

Доказательство. Согласно лемме 5.1 для любого плана $X = (x_1, x_2, \dots, x_n)$ задачи (5.1) — (5.3) справедливо неравенство

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i^*,$$

откуда, учитывая равенство (5.27), получаем

$$\sum_{j=1}^n c_j x_j \leq \sum_{j=1}^n c_j x_j^*. \quad (5.28)$$

Неравенство (5.28), справедливое для любого плана X задачи (5.1) — (5.3), указывает на оптимальность плана X^* . Оптимальность плана Y^* двойственной задачи (5.4), (5.5) устанавливается аналогично. Лемма 5.2 доказана.

Экономический смысл утверждения леммы 5.2 состоит в следующем. Если при некотором плане X^* использования способов производства и некотором векторе Y^* предварительных оценок факторов производства оценка произведенной продукции оказывается равной суммарной оценке имеющихся ресурсов, то X^* и Y^* являются соответственно оптимальным планом производства и вектором оценок факторов производства.

Лемма 5.2 устанавливает достаточность условия (5.27) для оптимальности планов X^* и Y^* . В дальнейшем мы

убедимся, что равенство (5.27) является также и необходимым условием для оптимальности планов X^* и Y^* .

Лемма 5.3. *Если линейная форма (5.4) двойственной задачи (5.4), (5.5) не ограничена снизу на множестве своих планов, то прямая задача (5.1) — (5.3) не имеет ни одного плана.*

Доказательство. По условию существует последовательность планов $\{Y_k\}$ двойственной задачи (5.4), (5.5) такая, что

$$\lim_{k \rightarrow \infty} (B, Y_k) = -\infty. \quad (5.29)$$

Если предположить, что прямая задача (5.1) — (5.3) имеет план X , то, согласно лемме 5.1,

$$(C, X) \leq (B, Y_k)$$

для любого натурального k .

Переходя в этом неравенстве к пределу при $k \rightarrow \infty$ и учитывая (5.29), получаем

$$(C, X) = -\infty. \quad (5.30)$$

Если все составляющие вектора X — конечные величины, то равенство (5.30) невозможно. Следовательно, предположение о наличии у задачи (5.1) — (5.3) хотя бы одного плана ошибочно. Лемма 5.3 доказана.

Поясним геометрический смысл утверждения леммы 5.3 в $(m+1)$ -мерном пространстве точек $U = (u_1, u_2, \dots, u_{m+1})$. Если существует последовательность гиперплоскостей, содержащих начало координат, расположенных «над» конусом K и пересекающих ось Q в точках, $(m+1)$ -я координата которых стремится к $-\infty$, то прямая Q и конус K не имеют ни одной общей точки.

Лемма 5.3 оказывается полезной при установлении неразрешимости задачи линейного программирования.

Все три леммы этого пункта формулировались для задачи линейного программирования, записанной в канонической форме. Естественно, что эти утверждения имеют место и для задач с однотипными условиями, поскольку при построении задачи, двойственной по отношению к задаче (5.17) — (5.19), мы приводили последнюю к кано-

нической форме. Впрочем, отмеченным обстоятельством мы уже пользовались при пояснении экономической сущности приведенных здесь предложений.

§ 6. ТЕОРЕМЫ ДВОЙСТВЕННОСТИ

6.1. Как было показано в § 5, каждая из задач линейного программирования с однотипными условиями (5.17)—(5.19) и (5.20)—(5.22) является двойственной по отношению к другой задаче. Таким образом, обе они составляют двойственную (взаимосопряженную) пару задач линейного программирования. Взаимосопряженные задачи линейного программирования обладают рядом интересных и важных для приложений свойств, связывающих их воедино. Некоторые простейшие свойства двойственных задач были уже рассмотрены в конце § 5. Настоящий параграф посвящен выяснению более глубоких связей между задачами двойственной пары. Вначале будут рассмотрены задачи с однотипными условиями. Общий случай исследуется далее.

6.2. Займемся изучением произвольной пары взаимосопряженных задач с однотипными условиями. Допустим, что эта пара состоит из задач (5.17)—(5.19) и (5.20)—(5.22). Сформулируем и докажем основное утверждение теории двойственности, которое устанавливает тесную связь между решениями задач двойственной пары.

Теорема 6.1 (первая теорема двойственности). *Если одна из задач двойственной пары (5.17)—(5.19) и (5.20)—(5.22) имеет решение, то другая задача также разрешима. При этом для любых оптимальных планов*

$$X = (x_1, x_2, \dots, x_n) \quad \text{и} \quad Y = (y_1, y_2, \dots, y_m)$$

задач (5.17)—(5.19) и (5.20)—(5.22) имеет место равенство

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i. \quad (6.1)$$

Доказательство. 1. Предположим, что разрешимой является задача (5.17)—(5.19). Рассуждения будут

проводиться в $(n+1)$ -мерном пространстве точек $\bar{X} = (x_1, x_2, \dots, x_n, x_{n+1})$.

Введем в рассмотрение выпуклый многогранный конус R с вершиной в точке $\bar{0} = (0, 0, \dots, 0)$, образованный векторами

$$\begin{aligned}\bar{A}_i &= (a_{i1}, a_{i2}, \dots, a_{in}, -b_i), \quad i = 1, 2, \dots, m, \\ -\bar{e}_i &= (\underbrace{0, 0, \dots, 0}_i, -1, 0, \dots, 0), \quad i = 1, 2, \dots, n+1.\end{aligned}$$

Другими словами, объединим в множество R полную совокупность векторов \bar{X} , представимых в виде

$$\bar{X} = \sum_{i=1}^m \lambda_i \bar{A}_i - \sum_{i=1}^{n+1} \mu_i \bar{e}_i, \quad \lambda_i \geq 0, \mu_i \geq 0. \quad (6.2)$$

Покажем, что точка

$$\bar{C} = (c_1, c_2, \dots, c_n, -L^*),$$

где L^* — максимальное значение линейной формы (5.17) при условиях (5.18), (5.19), принадлежит конусу R .

2. Предположим, что это не так, т. е. $\bar{C} \notin R$. Применим к выпуклому замкнутому множеству R в точке $\bar{C} \notin R$ следствие 1.2.

В соответствии с этим следствием существует гиперплоскость

$$(\bar{W}, \bar{X}) = \alpha, \quad (6.3)$$

обладающая следующими свойствами:

а) точка \bar{C} расположена внутри верхнего полупространства гиперплоскости (6.3), т. е.

$$(\bar{W}, \bar{C}) > \alpha; \quad (6.4)$$

б) гиперплоскость (6.3) является опорной для конуса R , причем R лежит в ее нижнем полупространстве, т. е.

$$\max_{\bar{X} \in R} (\bar{W}, \bar{X}) = \alpha. \quad (6.5)$$

Убедимся в том, что свободный член α в уравнении (6.3) гиперплоскости равен нулю. Действительно, поскольку

$0 \in R$, то, согласно (6.5),

$$\alpha \geq 0.$$

Пусть \bar{X}_0 — точка конуса R , которую содержит гиперплоскость (6.3):

$$(\bar{W}, \bar{X}_0) = \alpha.$$

Если допустить, что $\alpha > 0$, то для точки $\bar{X}_1 = 2\bar{X}_0 \in R$ получаем неравенство

$$(\bar{W}, \bar{X}_1) = 2\alpha > \alpha,$$

которое противоречит условию (6.5). Следовательно, $\alpha = 0$. Теперь неравенство (6.4) может быть переписано в виде

$$\sum_{j=1}^n w_j c_j - w_{n+1} L^* > 0, \quad (6.6)$$

а соотношение (6.5) с учетом представления (6.2) переходит в систему неравенств

$$\left. \begin{aligned} \sum_{j=1}^n w_j a_{ij} - w_{n+1} b_i &\leq 0, \quad i = 1, 2, \dots, m, \\ w_j &\geq 0, \quad j = 1, 2, \dots, n, n+1, \end{aligned} \right\} \quad (6.7)$$

где

$$\bar{W} = (w_1, w_2, \dots, w_n, w_{n+1}).$$

3. Покажем, что последняя компонента w_{n+1} вектора \bar{W} положительна. В самом деле, если допустить противное ($w_{n+1} = 0$), то соотношения (6.6) и (6.7) переходят соответственно в

$$\sum_{j=1}^n w_j c_j > 0, \quad (6.8)$$

и

$$\sum_{j=1}^n w_j a_{ij} \leq 0, \quad i = 1, 2, \dots, m, \quad (6.9)$$

$$w_j \geq 0, \quad j = 1, 2, \dots, n. \quad (6.10)$$

Пусть $X = (x_1, x_2, \dots, x_n)$ — произвольный план задачи (5.17)—(5.19). Согласно неравенствам (6.9) и (6.10) вектор

$$X(\lambda) = (x_1 + \lambda w_1, x_2 + \lambda w_2, \dots, x_n + \lambda w_n)$$

является планом задачи (5.17)—(5.19) при любом $\lambda \geq 0$. С другой стороны, неравенство (6.8) приводит к соотношению

$$\lim_{\lambda \rightarrow \infty} \sum_{j=1}^n (x_j + \lambda w_j) c_j = \infty,$$

которое противоречит разрешимости задачи (5.17)—(5.19). Следовательно,

$$w_{n+1} > 0. \quad (6.11)$$

Положим $z_j = w_j/w_{n+1}$ для $j = 1, 2, \dots, n$ и, учитывая (6.11), перепишем соотношения (6.6) и (6.7) в эквивалентном виде:

$$\sum_{j=1}^n z_j c_j > L^*, \quad (6.12)$$

$$\sum_{j=1}^n z_j a_{ij} \leq b_i, \quad i = 1, 2, \dots, m, \quad (6.13)$$

$$z_j \geq 0, \quad j = 1, 2, \dots, n. \quad (6.14)$$

Но, как нетрудно проверить, система неравенств (6.12)—(6.14) противоречива. Действительно, из соотношений (6.13), (6.14) следует, что вектор Z — план задачи (5.17)—(5.19), и поэтому должно выполняться неравенство

$$\sum_{j=1}^n z_j c_j \leq L^*,$$

которое противоречит неравенству (6.12).

Итак, предположение, что $\bar{C} \notin R$, привело к противоречию. Следовательно, $\bar{C} \in R$.

4. Поскольку точка \bar{C} содержится в многогранном конусе R , то для нее справедливо представление (6.2), т. е. найдутся такие неотрицательные числа:

$$y_1, y_2, \dots, y_m, \quad v_1, v_2, \dots, v_n,$$

что

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

$$-L^* = - \sum_{i=1}^m y_i b_i - v_{n+1}.$$

Переписывая полученные соотношения в более удобной форме, имеем

$$\left. \begin{aligned} c_j &\leq \sum_{i=1}^m y_i a_{ij}, \quad j = 1, 2, \dots, n, \\ y_i &\geq 0, \quad i = 1, 2, \dots, m, \end{aligned} \right\} \quad (6.15)$$

$$L^* \geq \sum_{i=1}^m y_i b_i. \quad (6.16)$$

Согласно (6.15), вектор $Y = (y_1, y_2, \dots, y_m)$ является планом задачи (5.20)—(5.22). Поэтому по лемме 5.1

$$\sum_{i=1}^m y_i b_i \geq L^*. \quad (6.17)$$

Сравнивая (6.16) и (6.17), получаем

$$L^* = \sum_{i=1}^m y_i b_i.$$

В соответствии с леммой 5.2 полученное равенство означает, что план

$$Y = (y_1, y_2, \dots, y_m)$$

является решением задачи (5.20)—(5.22).

Итак, предположив задачу (5.17)—(5.19) разрешимой, мы доказали существование оптимального плана у двойственной задачи (5.20)—(5.22) и установили совпадение оптимальных значений линейных форм обеих задач. Если учесть, что задача (5.17)—(5.19) является двойственной по отношению к задаче (5.20)—(5.22), то доказательство разрешимости задачи (5.17)—(5.19) в предположении разрешимости задачи (5.20)—(5.22) осуществляется точно так же.

Первая теорема двойственности доказана.

63. Приведем несколько следствий из первой теоремы двойственности.

Следствие 6.1. Для разрешимости одной из задач двойственной пары (5.17)—(5.19) и (5.20)—(5.22) необходимо и достаточно, чтобы каждая из этих задач имела хотя бы один план.

Доказательство. а) Достаточность сформулированного утверждения устанавливается без применения первой теоремы двойственности.

Пусть $Y' = (y'_1, y'_2, \dots, y'_m)$ — произвольный план задачи (5.20)—(5.22). В таком случае, согласно лемме 5.1, для любого плана $X = (x_1, x_2, \dots, x_n)$ задачи (5.17)—(5.19) справедливо неравенство

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y'_i.$$

Итак, множество планов задачи (5.17)—(5.19) не пусто, и линейная форма (5.17) ограничена на нем сверху. Поэтому в соответствии с теоремой 3.4 задача (5.17)—(5.19) разрешима.

Разрешимость задачи (5.20)—(5.22) устанавливается аналогично.

б) Необходимость указанных условий следует из первой теоремы двойственности. Действительно, если одна из задач двойственной пары разрешима, то разрешима и другая задача. Следовательно, каждая из задач двойственной пары имеет по крайней мере один план.

Следствие 6.2. Для того чтобы одна из задач двойственной пары имела планы, а множество планов другой задачи было пусто, необходима и достаточна неограниченность линейной формы первой задачи на множестве ее планов.

Доказательство. а) Достаточность условий следствия 6.2 составляет содержание леммы 5.3.

б) Для установления необходимости указанных условий обратимся к первой теореме двойственности. Если множество планов одной из задач двойственной пары пусто, то эта задача неразрешима и, следовательно, другая задача также не имеет решения. Но по условию последняя задача имеет планы. Следовательно, ее нераз-

решимость обусловлена неограниченностью линейной формы на множестве планов (см. теорему 3.4).

При исследовании задач двойственной пары можно столкнуться с одной из трех взаимно исключающих друг друга возможностей:

- а) обе задачи имеют планы;
- б) планы имеются только у одной задачи;
- в) для каждой задачи двойственной пары множество планов пусто.

Приведем соответствующие примеры, показывающие, что каждая из отмеченных возможностей а), б) и в) действительно реализуема.

Пример 1. Задача I':

$$5x_1 + x_2 \rightarrow \max,$$

$$x_1 + 2x_2 \leq 4,$$

$$2x_1 + 3x_2 \leq 3,$$

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

Задача II', двойственная по отношению к задаче I', имеет вид

$$4y_1 + 3y_2 \rightarrow \min,$$

$$y_1 + 2y_2 \geq 5,$$

$$2y_1 + 3y_2 \geq 1,$$

$$y_1 \geq 0, \quad y_2 \geq 0.$$

Очевидно, каждая из этих задач имеет планы:

вектор $(0, 0)$ — план задачи I',

вектор $(1, 2)$ — план задачи II'.

Пример 2. Прямая задача I'' имеет вид:

$$5x_1 + x_2 \rightarrow \max,$$

$$x_1 - x_2 \leq 2,$$

$$x_1 - 3x_2 \leq 3,$$

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

Двойственная задача II'' выглядит следующим образом:

$$2y_1 + 3y_2 \rightarrow \min,$$

$$y_1 + y_2 \geq 5,$$

$$-2y_1 - 3y_2 \geq 1,$$

$$y_1 \geq 0, \quad y_2 \geq 0.$$

Вектор $(0, 0)$ является планом задачи I''.

Задача II'' не имеет ни одного плана. В самом деле, умножая первое условие задачи II'' на 2 и складывая со вторым, имеем

$$-y_2 \geq 11,$$

или

$$y_2 \leq -11,$$

что противоречит последнему условию задачи: $y_2 \geq 0$.

Пример 3. Прямая задача I''':

$$5x_1 + x_2 \rightarrow \max,$$

$$x_1 - x_2 \leq 1,$$

$$-x_1 + x_2 \leq -2,$$

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

Двойственная задача II''':

$$y_1 - 2y_2 \rightarrow \min,$$

$$y_1 - y_2 \geq 5,$$

$$-y_1 + y_2 \geq 1,$$

$$y_1 \geq 0, \quad y_2 \geq 0.$$

Сложим два первых условия задачи I''':

$$0 \leq -1.$$

Полученное противоречие указывает на отсутствие планов у задачи I'''. Прделав ту же самую операцию над условиями задачи II''', получаем $0 \geq 6$. Следовательно, задача II''' также не имеет ни одного плана.

Из следствия 6.1 вытекает, что условия случая а) эквивалентны предположению о разрешимости обеих задач двойственной пары. Обратившись к следствию 6.2, видим, что случай б) равносильен требованию неограниченности линейной формы одной из задач на множестве ее планов.

Приведем еще одно следствие из первой теоремы двойственности, содержащее необходимое и достаточное условие оптимальности планов взаимосопряженных задач.

Следствие 6.3. Для оптимальности планов $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ и $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$ задач (5.17)–(5.19) и (5.20)–(5.22) соответственно необходимо и достаточно выполнение равенства

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*.$$

Доказательство. Достаточность этого условия следует из леммы 5.2.

Необходимость составляет содержание второй части первой теоремы двойственности.

6.4. Для дальнейшего нам понадобится несколько новых определений. Рассмотрим условия взаимосопреженных задач (5.17)—(5.19) и (5.20)—(5.22). Условия систем (5.19), (5.21)

$$x_j \geq 0, \quad \sum_{i=1}^m a_{ij} y_i \geq c_j \quad (6.18)$$

при фиксированном значении индекса j и условия систем (5.18), (5.22)

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad y_i \geq 0 \quad (6.19)$$

при фиксированном значении индекса i будем называть *парами двойственных условий* взаимосопреженных задач.

Двойственная пара (6.18) отвечает j -му столбцу таблицы 3.1, и поэтому составляющие ее условия называются столбцовыми. По той же причине условия двойственной пары (6.19) называются строчными (они соответствуют i -й строке таблицы 3.1).

Условие задачи (5.17)—(5.19) или задачи (5.20)—(5.22) назовем *закрепленным*, если на *любом* оптимальном плане соответствующей задачи оно удовлетворяется как точное равенство.

Условие задачи (5.17)—(5.19) или (5.20)—(5.22) назовем *свободным*, если *хотя бы для одного* оптимального плана соответствующей задачи оно выполняется как строгое неравенство.

Формулируемая ниже теорема устанавливает связь между условиями произвольной двойственной пары.

Теорема 6.2 (вторая теорема двойственности). *Если взаимосопреженные задачи (5.17)—(5.19) и (5.20)—(5.22) разрешимы, то в каждой паре их двойственных условий одно условие свободное, а другое закрепленное.*

Доказательство этой теоремы мы приводить не будем, а попытаемся более отчетливо представить себе содержание обеих теорем двойственности.

Утверждения теорем двойственности становятся особенно прозрачными, если переформулировать их в терминах производства однородного продукта. Эти термины уже использовались для экономической интерпретации пары двойственных задач с однотипными условиями. Согласно первой теореме двойственности оптимальный план производства существует в том и только в том случае, если все факторы производства имеют оценки. Допустим, что производство обладает оптимальным планом и, следовательно, все его факторы могут быть оценены. Следует заметить, что в реальных случаях это всегда так. Вообще говоря, как оптимальный план производства, так и система оценок факторов производства определяются неоднозначно.

Согласно второй части первой теоремы двойственности при любых оценках производственных факторов (составляющих решения двойственной задачи) оценка продукта, полученного реализацией любого оптимального плана производства, совпадает с суммарной оценкой имеющихся ресурсов. Таким образом, характеристическое свойство оптимального плана состоит в совпадении с точки зрения принятых оценок результата производства и его затрат. При любом другом плане использования технологических способов (отличном от оптимального) производство будет убыточным: оценка произведенной продукции окажется меньше, чем суммарная оценка имеющихся ресурсов. Это объясняется тем, что при неоптимальном плане возможности производства используются не полностью.

Напомним, что в качестве вектора оценок факторов производства выбирается такой набор предварительных оценок производственных факторов (план двойственной задачи), который обращает суммарную оценку имеющихся ресурсов (значение линейной формы двойственной задачи) в минимум. Смысл последнего условия становится теперь совершенно ясным. При соблюдении этого условия выполняется характеристическое свойство оптимального плана. Если же в качестве вектора оценок принять некоторый набор предварительных оценок, не обращающий в минимум суммарную оценку имеющихся ресурсов, то характеристическое свойство оптимального плана нарушается. В этом случае оценка результата производства,

работающего по любому плану, в том числе и по оптимальному, оказывается меньше суммарной оценки имеющихся в наличии ресурсов. Естественно, что оценки, при которых имеет место подобное положение, нельзя считать естественными.

Перейдем к экономической интерпретации второй теоремы двойственности. Вначале обратимся к строчным условиям двойственной пары задач, определяющих данное производство. Пусть при одном из оптимальных планов использования способов производства i -й производственный фактор используется не полностью. Тогда по второй теореме двойственности его оценка равна нулю. Наоборот, если оценка i -го производственного фактора (при любом векторе оценок) равна нулю, то существует такой оптимальный план производства, при котором ресурсы i -го фактора используются не в полном объеме.

Такое положение следует признать естественным. В самом деле, фактор, запасы которого превышают потребности в нем (с точки зрения некоторого оптимального плана производства), не представляет ценности для производства: некоторое сокращение запасов по такому фактору не уменьшит возможностей производства. Поэтому оценку данного фактора с точки зрения рассматриваемого производства естественно принять равной нулю.

Допустим теперь, что при любом оптимальном плане использования способов производства i -й производственный фактор расходуется полностью. Согласно второй теореме двойственности это влечет за собой положительность оценки i -го фактора. Наоборот, положительная оценка возможна только у такого фактора, который расходуется полностью при любом оптимальном плане использования технологических способов производства.

Приведем экономическое истолкование столбцовых условий рассматриваемой пары двойственных задач. Согласно второй теореме двойственности, данный способ производства используется в некотором оптимальном плане в том и только в том случае, если при его реализации оценки полученной продукции и затраченных ресурсов совпадают (с точки зрения любого вектора оценок).

Интуитивно это утверждение представляется вполне естественным: если некоторый способ производства связан

с превышением расходов над доходами, то его использование не имеет смысла.

6.5. В предыдущих пунктах предполагалось, что исходная задача имеет только однотипные условия.

Рассмотрим общий случай. Требуется обратить в максимум линейную форму

$$L(X) = \sum_{j=1}^n c_j x_j \quad (6.20)$$

при условиях

$$\left. \begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = 1, 2, \dots, m_1 \leq m, \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = m_1 + 1, m_1 + 2, \dots, m, \end{aligned} \right\} \quad (6.21)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n_1 \leq n. \quad (6.22)$$

По определению задача, двойственная по отношению к задаче (6.20) — (6.22) (или сопряженная с ней), состоит в минимизации линейной формы

$$\sum_{i=1}^{m^*} b_i y_i \quad (6.23)$$

при условиях

$$\left. \begin{aligned} \sum_{i=1}^m a_{ij} y_i &\geq c_j, \quad j = 1, 2, \dots, n_1 \leq n, \\ \sum_{i=1}^m a_{ij} y_i &= c_j, \quad j = n_1 + 1, n_1 + 2, \dots, n, \end{aligned} \right\} \quad (6.24)$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m_1 \leq m. \quad (6.25)$$

Таким образом, задача, сопряженная с задачей со смешанными условиями (6.20) — (6.22), составляется согласно следующим правилам. Если переменная x_j задачи (6.20) — (6.22) предполагается неотрицательной, то j -е условие системы (6.24) является неравенством. Если же на x_j подобное ограничение не накладывается, то j -е соотношение (6.24) представляет собой равенство. Аналогичные связи имеются между условиями (6.21) задачи (6.20) — (6.22) и ограничениями (6.25) задачи (6.23) —

(6.25). Если i -е условие системы (6.21) — неравенство, то $y_i \geq 0$. В противном случае переменная y_i может принимать как положительные, так и отрицательные значения.

Как нетрудно проверить, задача (6.20) — (6.22) является двойственной по отношению к задаче (6.23) — (6.25). Поэтому обе эти задачи имеет смысл называть парой двойственных (или взаимосопряженных) задач.

Заметим, что приведенное только что определение двойственной задачи не противоречит понятию двойственности, данному ранее для задач с однотипными условиями и задач в канонической форме. Если положить $m_1 = m$, $n_1 = n$, то задача (6.20) — (6.22) превращается в задачу с однотипными условиями (5.17) — (5.19), а двойственная задача (6.23) — (6.25) переходит в задачу (5.20) — (5.22). При $m_1 = 0$, $n_1 = n$ задачи (6.20) — (6.22) и (6.23) — (6.25) превращаются в задачи (5.1) — (5.3) и (5.4) — (5.5) соответственно.

Каждую задачу линейного программирования со смешанными условиями можно привести к эквивалентному виду, представляющему собой задачу с однотипными условиями. При этом задачи с однотипными условиями, эквивалентные задачам (6.20) — (6.22) и (6.23) — (6.24), составляют двойственную пару. Мы не будем останавливаться на этом подробнее, а заметим лишь, что указанный прием позволяет без труда перенести утверждения обеих теорем двойственности на общий случай. Формулировка первой теоремы двойственности сохраняется полностью; формулировка второй теоремы двойственности также не меняется, но относится лишь к условиям типа неравенств (к столбцовым условиям с номерами $1, 2, \dots, n_1$ и к строчным условиям с номерами $1, 2, \dots, m_1$).

§ 7. КРИТЕРИИ ОПТИМАЛЬНОСТИ И РАЗРЕШАЮЩИЕ МНОЖИТЕЛИ

7.1. При решении задач линейного программирования чрезвычайно важно знать способы, позволяющие проверять планы задачи на оптимальность. Другими словами, необходимо уметь ответить на вопрос, является ли данный план оптимальным или нет. Условия, необходимые и достаточные для оптимальности плана задачи линейного

программирования, впервые были найдены Л. В. Канторовичем [18]. В дальнейшем выяснилось, что эти условия (их принято называть критерием оптимальности плана соответствующей задачи) весьма тесно связаны с теорией двойственности и по существу являются следствием теорем двойственности. Изложение настоящего параграфа во многом основано на этой связи. Такой подход позволяет использовать результаты предшествующих параграфов данной главы и, как нам представляется, делает изложение более естественным.

Рассмотрим произвольную задачу со смешанными условиями (задача (6.20) — (6.22)). Назовем, следуя Л. В. Канторовичу, величины $\lambda_1, \lambda_2, \dots, \lambda_m$ *разрешающими множителями* *), если:

$$\text{а) } \sum_{i=1}^m a_{ij} \lambda_i \geq c_j, \quad j = 1, 2, \dots, n_1; \quad (7.1)$$

$$\text{б) } \sum_{i=1}^m a_{ij} \lambda_i = c_j, \quad j = n_1 + 1, n_1 + 2, \dots, n; \quad (7.2)$$

$$\text{в) } \lambda_i \geq 0, \quad i = 1, 2, \dots, m_1; \quad (7.3)$$

г) для некоторого плана $X = (x_1, x_2, \dots, x_n)$ задачи (6.20) — (6.22) выполняются условия

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j \quad \text{при} \quad x_j > 0 \quad (1 \leq j \leq n_1), \quad (7.4)$$

$$\lambda_i = 0 \quad \text{при} \quad \sum_{j=1}^n a_{ij} x_j < b_i \quad (1 \leq i \leq m_1). \quad (7.5)$$

Вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, компонентами которого являются разрешающие множители λ_i , назовем *разрешающим вектором задачи* (6.20) — (6.22) (*разрешающим вектором плана* X). Приводимое ниже утверждение показывает, что отыскание разрешающего вектора эквивалентно решению задачи (6.23) — (6.25), двойственной по отношению к задаче (6.20) — (6.22).

*) В книге Л. В. Канторовича [20] разрешающие множители именуются объективно обусловленными оценками. Этот термин связан с экономической интерпретацией решения сопряженной задачи.

Теорема 7.1. Совокупность разрешающих векторов задачи (6.20) — (6.22) совпадает с множеством оптимальных планов задачи (6.23) — (6.25).

Доказательство. 1. Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — произвольный разрешающий вектор задачи (6.20) — (6.23), связанный соотношениями (7.4), (7.5) с планом $X = (x_1, x_2, \dots, x_n)$ этой задачи.

Условия (7.1) — (7.3), которым удовлетворяет вектор Λ , показывают, что Λ — план задачи (6.23) — (6.25). Для установления оптимальности плана Λ проведем несложные выкладки

$$\sum_{j=1}^n c_j x_j = \sum_{j \in E} c_j x_j + \sum_{j=n_1+1}^n c_j x_j. \quad (7.6)$$

Здесь через E обозначена совокупность индексов j ($j = 1, 2, \dots, n_1$), для которых $x_j > 0$. Учитывая равенство (7.6), условия (7.2) при $j = n_1 + 1, \dots, n$ и (7.4) при $j \in E$, получаем

$$\sum_{j=1}^n c_j x_j = \sum_{j \in E} \left(\sum_{i=1}^m a_{ij} \lambda_i \right) x_j + \sum_{j=n_1+1}^n \left(\sum_{i=1}^m a_{ij} \lambda_i \right) x_j.$$

Но, по предположению, $x_j = 0$, если $j \notin E$, $j \leq n_1$. Следовательно,

$$\sum_{j=1}^n c_j x_j = \sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} \lambda_i \right) x_j = \sum_{i=1}^m \lambda_i \sum_{j=1}^n a_{ij} x_j. \quad (7.7)$$

Принимая, далее, во внимание равенство

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad \text{при } m_1 + 1 \leq i \leq m$$

и условие (7.5), имеем

$$\sum_{i=1}^m \lambda_i \sum_{j=1}^n a_{ij} x_j = \sum_{i=1}^m \lambda_i b_i.$$

Сравнение последнего равенства с равенством (7.7) приводит к соотношению

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i \lambda_i. \quad (7.8)$$

Согласно лемме 5.2 равенств (7.8) указывает на оптимальность планов X и Λ .

Итак, $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — решение задачи (6.23) — (6.25).

2. Пусть теперь $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — произвольное решение задачи (6.23) — (6.25). Поскольку вектор Λ — план задачи (6.23) — (6.25), то для него выполнены условия (7.1) — (7.3). Обозначим через E_1 совокупность номеров свободных условий системы (6.21), а через E_2 — аналогичное множество индексов системы (6.22).

В соответствии со второй теоремой двойственности i -е условие системы (6.25) (j -е условие системы (6.24)) является закрепленным, если $i \in E_1$ ($j \in E_2$). Следовательно,

$$\lambda_i = 0 \quad \text{при} \quad i \in E_1, \quad (7.9)$$

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j \quad \text{при} \quad j \in E_2. \quad (7.10)$$

Рассмотрим произвольное решение $X = (x_1, x_2, \dots, x_n)$ задачи (6.20) — (6.22). Если $x_j > 0$, то, по определению, $j \in E_2$, и имеет место соотношение (7.10). Если $\sum_{j=1}^n a_{ij} x_j < b_i$, то $i \in E_1$, и выполняется равенство (7.9).

Таким образом, вектор Λ удовлетворяет условиям (7.1) — (7.3) и связан соотношениями (7.4), (7.5) с некоторым планом задачи (6.20) — (6.22) (в качестве этого плана можно принять любое решение задачи). Следовательно, $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — разрешающий вектор задачи (6.20) — (6.22). Теорема доказана.

7.2. В терминах разрешающих множителей удобно формулировать условия оптимальности планов задач линейного программирования.

Теорема 7.2 (критерий оптимальности плана задачи (6.20) — (6.22)). Для оптимальности плана $X = (x_1, x_2, \dots, x_n)$ задачи (6.20) — (6.22) необходимо и достаточно существование разрешающего вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, связанного с этим планом условиями (7.4) — (7.5).

Доказательство. 1. Необходимость. Предположим, что $X = (x_1, x_2, \dots, x_n)$ — решение задачи

(6.20) — (6.22). В силу первой теоремы двойственности сопряженная задача (6.23) — (6.25) разрешима.

Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — один из оптимальных планов этой задачи. В таком случае, согласно теореме 7.1, вектор Λ является разрешающим вектором задачи (6.20) — (6.22), причем, как было показано при доказательстве второй части теоремы 7.1, вектор Λ связан соотношениями (7.4), (7.5) с любым решением задачи (6.20) — (6.22), а следовательно, и с рассматриваемым решением X .

2. Достаточность. Допустим, что существует разрешающий вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, связанный с данным планом X задачи (6.20) — (6.22) условиями (7.4), (7.5).

В процессе доказательства первой части теоремы 7.1 было установлено равенство (7.8). Из этого равенства (если учесть, что X и Λ являются планами задач (6.20) — (6.22) и (6.23) — (6.25) соответственно) вытекает оптимальность плана X (см. лемму 5.2). Теорема доказана.

Установленный критерий позволяет сравнительно просто выяснить, является ли данный план решением задачи или нет. Общая схема проверки оптимальности плана состоит в следующем.

Пользуясь системой, составленной из уравнений (7.2), (7.4), (7.5), определяют вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$. Затем непосредственной подстановкой проверяют, удовлетворяет этот вектор условиям (7.1), (7.3) или нет. В первом случае вектор X — решение задачи; во втором — план, не являющийся оптимальным.

Обычно критерий оптимальности используется при анализе опорных планов задачи. Остановимся на этом подробнее. Рассмотрим некоторый опорный план X задачи (6.20) — (6.22). Допустим, что

$$\left. \begin{aligned} x_j &= 0 \quad \text{при} \quad j = 1, 2, \dots, n_2 \leq n_1, \\ x_j &> 0 \quad \text{при} \quad j = n_2 + 1, \dots, n_1, \end{aligned} \right\} \quad (7.11)$$

$$\left. \begin{aligned} \sum_{j=1}^n a_{ij} x_j &< b_i \quad \text{при} \quad i = 1, 2, \dots, m_2 \leq m_1, \\ \sum_{j=1}^n a_{ij} x_j &= b_i \quad \text{при} \quad i = m_2 + 1, \dots, m_1. \end{aligned} \right\} \quad (7.12)$$

Выпишем матрицу, составленную из коэффициентов условий задачи, которые обращаются планом X в точные равенства:

$$A_X = \left\{ \begin{array}{cccccc} 1 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 0 \\ a_{m_2+1,1} & a_{m_2+1,2} & \dots & a_{m_2+1,n_2} & \dots & a_{m_2+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m_1} & a_{m_2} & \dots & a_{mn_1} & \dots & a_{mn} \end{array} \right\} \begin{array}{l} n_2 \\ m - m_2 \end{array}$$

Поскольку план X по условию опорный, то среди строк матрицы A_X (их общее число $m + n_2 - m_2$) имеется n линейно независимых.

Единичные векторы, составляющие первые n_2 строк матрицы A_X , линейно независимы. Поэтому из числа остальных $m - m_2$ строк можно выделить такие $n - n_2$ строк, которые вместе с первыми n_2 строками матрицы A_X составят линейно независимую систему.

Образует из полученной линейно независимой системы строк определитель порядка n и разложим его по первым n_2 строкам. В результате получим отличный от нуля определитель порядка $n - n_2$, строками которого являются некоторые из векторов системы

$$(a_{i, n_2+1}, a_{i, n_2+2}, \dots, a_{i, n}), \quad i = m_2 + 1, \dots, m. \quad (7.13)$$

Следовательно, система $(n - n_2)$ -мерных векторов (7.13) имеет ранг, равный $n - n_2$.

Если допустить, что план X является невырожденным, то число строк матрицы A_X равняется числу ее столбцов, т. е. $m - m_2 + n_2 = n$. В этом случае число векторов системы (7.13), равное $m - m_2$, совпадает с ее рангом $n - n_2$. Итак, если план X невырожденный, то связанная с ним система (7.13) состоит из $n - n_2$ линейно независимых $(n - n_2)$ -мерных векторов. Если же план X оказывается вырожденным, то число векторов системы (7.13) превышает их размерность.

После этих предварительных замечаний приступим к исследованию плана X на оптимальность. Для этого попытаемся подобрать разрешающий вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, связанный с данным планом условиями (7.4) — (7.5).

Учитывая условие (7.4) и предположение (7.11), получаем

$$\sum_{i=1}^m \lambda_i a_{ij} = c_j \quad \text{для } j = n_2 + 1, n_2 + 2, \dots, n.$$

С другой стороны, согласно условию (7.5) и предположению (7.12)

$$\lambda_i = 0 \quad \text{для } i = 1, 2, \dots, m_2.$$

Следовательно, остальные $m - m_2$ компонент разрешающего вектора Λ (если таковой существует) обязаны удовлетворять системе уравнений

$$\sum_{i=m_2+1}^m \lambda_i a_{ij} = c_j, \quad j = n_2 + 1, \dots, n. \quad (7.14)$$

Столбцами матрицы коэффициентов системы (7.14) являются векторы (7.13). Поэтому ранг системы (7.14) равен $n - n_2$, т. е. числу уравнений этой системы.

Анализ плана X особенно прост, если этот план оказывается невырожденным. В этом случае число неизвестных системы (7.14) равно числу ее уравнений, и система имеет единственное решение. Разрешив систему, получаем единственный вектор Λ , удовлетворяющий условиям (7.2), (7.4), (7.5). Подставим вектор Λ в левую часть соотношений (7.1), (7.3), которые в данном случае имеют вид

$$\left. \begin{aligned} \sum_{i=m_2+1}^m a_{ij} \lambda_i &\geq c_j, \quad j = 1, 2, \dots, n_2, \\ \lambda_i &\geq 0, \quad i = m_2 + 1, \dots, m_1. \end{aligned} \right\} \quad (7.15)$$

Если все условия (7.15) удовлетворятся, то Λ является разрешающим вектором. Согласно критерию оптимальности это означает, что план X — решение рассматриваемой задачи.

Если же хотя бы одно из соотношений (7.15) окажется нарушенным, то это будет означать, что разрешающего вектора для плана X не существует и, следовательно, этот план не является оптимальным. Заметим, что последнее утверждение является следствием единственности решения системы (7.14).

Дело обстоит сложнее, если план X оказывается вырожденным. В этом случае число неизвестных системы (7.14) превышает число ее уравнений. Поэтому система (7.14) имеет бесчисленное множество решений. Согласно критерию для оптимальности плана X необходимо и достаточно, чтобы хотя бы одно решение системы (7.14) удовлетворяло условиям (7.15).

Разрешим систему (7.14) относительно некоторых $n - n_2$ неизвестных и полученные выражения подставим в условия (7.1), (7.3). В результате образуется система T , состоящая из $t = m_1 - m_2 + n_2$ неравенств, связывающих $s = m - m_2 + n_2 - n$ переменных. Оптимальность плана X эквивалентна разрешимости полученной системы неравенств. Очевидно, ранг системы T равен s . В случае разрешимости системы неравенств T совокупность ее решений образует многогранное множество, имеющее вершины.

Следовательно, для выяснения разрешимости T достаточно найти решения всех систем уравнений, составленных из s линейно независимых условий, входящих в T . Число таких систем, очевидно, не превышает C_t^s . Если хотя бы одно из полученных решений удовлетворяет остальным соотношениям системы T , то данная система неравенств разрешима и, следовательно, исследуемый план X является оптимальным. В противном случае система неравенств T не имеет ни одного решения, что указывает на неоптимальность плана X .

При небольших значениях s и C_t^s предложенный путь выяснения разрешимости системы T можно считать приемлемым. Однако при достаточно больших величинах этих параметров его реализация связана с огромной вычислительной работой. В связи с этим в методах линейного программирования указанный путь применяется в несколько усовершенствованном виде:

а) процесс перехода от одной системы s уравнений с s неизвестными к другой системе упорядочивается, так что анализ всех таких систем обычно оказывается излишним;

б) каждый переход осуществляется по простым рекуррентным формулам.

Резюмируя все сказанное, можно сделать следующие выводы:

1. Практическое применение критерия оптимальности в случае невырожденности исследуемого опорного плана сводится к решению одной системы линейных уравнений.

2. В вырожденном случае применение критерия оптимальности связано с исследованием системы неравенств, что эквивалентно решению нескольких систем линейных уравнений.

7.3. Из теоремы 7.2 следует, что разрешающий вектор может быть связан условиями (7.4), (7.5) только с оптимальным планом задачи (6.20) — (6.22). Вместе с тем для каждого оптимального плана X рассматриваемой задачи существует свой разрешающий вектор, связанный с ним соотношениями (7.4), (7.5). Возникает вопрос, существует ли зависимость между разрешающим вектором некоторого оптимального плана задачи (6.20) — (6.22) и произвольным решением этой задачи? Ответом на этот вопрос служит следующее утверждение.

Теорема 7.3 (теорема о разрешающих векторах). *Разрешающий вектор некоторого оптимального плана задачи (6.20) — (6.22) является разрешающим и для любого другого решения этой задачи.*

Доказательство. Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — разрешающий вектор плана X , т. е. разрешающий вектор задачи (6.20) — (6.22), связанный с X условиями (7.4), (7.5). В соответствии с теоремой 7.1 вектор Λ является решением двойственной задачи (6.23) — (6.25). При доказательстве второй части теоремы 7.1 было установлено, что любое решение двойственной задачи связано условиями (7.4), (7.5) с произвольным решением прямой задачи. Следовательно, Λ является разрешающим вектором для любого оптимального плана задачи (6.20) — (6.22). Теорема доказана.

Доказанное утверждение дает основание связывать понятие разрешающего вектора с полным множеством оптимальных планов данной задачи. Поэтому можно говорить о разрешающем векторе задачи линейного программирования, не указывая, с каким планом этот вектор связан.

Подчеркнем еще раз, что вычисление разрешающего вектора задачи (разрешающих множителей) в общем

случае несколько не проще, чем решение самой задачи. Как мы видели, определение разрешающего вектора задачи эквивалентно решению задачи, двойственной по отношению к данной. Основная роль разрешающих множителей состоит в том, что в терминах этих множителей удобно формулировать критерии оптимальности, широко используемые в вычислительных методах и теоретических приложениях линейного программирования.

При описании методов линейного программирования мы обычно будем иметь дело с канонической формой задач. Поэтому целесообразно сформулировать критерий оптимальности для этого класса задач линейного программирования.

Итак, рассмотрим задачу (5.1) — (5.3), которая, очевидно, является частным случаем задачи (6.20) — (6.22) при $m_1 = 0$, $n_1 = n$. Определение разрешающего вектора для данной задачи принимает следующий вид. Вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ называется *разрешающим вектором* задачи (5.1) — (5.3), если:

$$a) \quad \sum_{i=1}^m a_{ij} \lambda_i \geq c_j, \quad j = 1, 2, \dots, n;$$

б) для некоторого плана $X = (x_1, x_2, \dots, x_n)$ задачи (5.1) — (5.3) выполняется условие

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j, \quad \text{если } x_j > 0. \quad (7.16)$$

Теорема 7.4 (критерий оптимальности плана задачи (5.1) — (5.3)). *Для оптимальности плана X задачи (5.1) — (5.3) необходимо и достаточно существование разрешающего вектора Λ , связанного с X условием (7.16).*

Сформулированная теорема является частным случаем общего критерия оптимальности (теорема 7.2). Практическое использование указанного критерия осуществляется согласно замечаниям предыдущего пункта.

7.4. При решении условных экстремальных задач классического анализа обычно используется метод Лагранжа. Напомним вкратце содержание этого метода.

Пусть требуется максимизировать или минимизировать функцию

$$\mathcal{F}(X) = \mathcal{F}(x_1, x_2, \dots, x_n), \quad (7.17)$$

переменные которой связаны условиями

$$G_i(X) = G_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, 2, \dots, m \quad (m < n). \quad (7.18)$$

Предположим, что функции $\mathcal{F}(X)$ и $G_i(X)$, $i = 1, 2, \dots, m$, непрерывны и обладают непрерывными частными производными первого порядка по всем переменным. Будем говорить, что система уравнений (7.18) *регулярна* в точке $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$, если определитель матрицы

$$\begin{vmatrix} \frac{\partial G_1(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_1(\bar{X})}{\partial x_{i_2}} & \dots & \frac{\partial G_1(\bar{X})}{\partial x_{i_m}} \\ \frac{\partial G_2(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_2(\bar{X})}{\partial x_{i_2}} & \dots & \frac{\partial G_2(\bar{X})}{\partial x_{i_m}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial G_m(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_m(\bar{X})}{\partial x_{i_2}} & \dots & \frac{\partial G_m(\bar{X})}{\partial x_{i_m}} \end{vmatrix}$$

не равен нулю. Здесь i_1, i_2, \dots, i_m — произвольные m индексов из системы индексов $(1, 2, \dots, n)$.

Метод Лагранжа основывается на следующем утверждении.

Если функция (7.17) достигает своего максимума или минимума при условиях (7.18) в точке \bar{X} и система (7.18) регулярна в этой точке, то существуют такие числа $\lambda_1, \lambda_2, \dots, \lambda_m$, что для функции

$$F_\Lambda(X) = \mathcal{F}(X) + \sum_{i=1}^m \lambda_i G_i(X) \quad (7.19)$$

в точке \bar{X} выполняются необходимые условия безусловного экстремума, т. е.

$$\frac{\partial F_\Lambda(\bar{X})}{\partial x_i} = 0 \quad \text{при} \quad i = 1, 2, \dots, n.$$

Числа $\lambda_1, \lambda_2, \dots, \lambda_m$ принято называть *множителями Лагранжа*, функцию $F_\Lambda(X)$ — *функцией Лагранжа*. Таким образом, вычисление условного экстремума (7.17), (7.18) сводится к отысканию безусловного экстремума функции Лагранжа (7.19).

Общая схема метода Лагранжа состоит в следующем. Составляется функция Лагранжа с неопределенными множителями λ_i . Затем решается система уравнений

$$\frac{\partial F_\Lambda(X)}{\partial x_i} = 0, \quad i = 1, 2, \dots, n.$$

Решение этой системы зависит от значений неизвестных параметров $\lambda_i, i = 1, 2, \dots, m$, которые определяются с помощью системы (7.18).

Задача математического программирования отличается от классической задачи на условный экстремум наличием условий, имеющих вид неравенств. Поэтому приведенный здесь метод Лагранжа к ней неприменим. Однако после некоторого видоизменения этот метод может быть распространен также и на достаточно широкий класс задач математического программирования. Здесь мы рассмотрим лишь случай линейного программирования.

Пусть задача линейного программирования (5.1) — (5.3) записана в канонической форме. Положим

$$\mathcal{F}(X) = \sum_{j=1}^n c_j x_j, \quad G_i(X) = - \sum_{j=1}^n a_{ij} x_j + b_i, \quad i = 1, 2, \dots, m.$$

Теорема 7.5. Для оптимальности плана \bar{X} задачи (5.1) — (5.3) необходимо и достаточно, чтобы функция Лагранжа

$$F_\Lambda(X) = \mathcal{F}(X) + \sum_{i=1}^m \lambda_i G_i(X)$$

при некоторых значениях множителей $\lambda_i (i = 1, 2, \dots, m)$ достигала в точке \bar{X} максимума при условии $x_j \geq 0, j = 1, 2, \dots, n$.

Доказательство. Необходимость. Пусть $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ — оптимальный план задачи (5.1) —

(5.3). Примем в качестве вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ произвольный разрешающий вектор задачи. Введем

$$\begin{aligned} F_{\Lambda}(X) &= \sum_{j=1}^n c_j x_j + \sum_{i=1}^m \lambda_i \left(- \sum_{j=1}^n a_{ij} x_j + b_i \right) = \\ &= \sum_{j=1}^n x_j \left(c_j - \sum_{i=1}^m \lambda_i a_{ij} \right) + \sum_{i=1}^m \lambda_i b_i. \end{aligned}$$

В соответствии с определением разрешающего вектора

$$c_j - \sum_{i=1}^m \lambda_i a_{ij} \leq 0 \quad \text{для всех } j = 1, 2, \dots, n, \quad (7.20)$$

причем

$$c_j - \sum_{i=1}^m \lambda_i a_{ij} = 0, \quad (7.21)$$

если $\bar{x}_j > 0$. Учитывая (7.21), можно переписать выражение для $F_{\Lambda}(X)$ в виде

$$F_{\Lambda}(X) = \sum_{j \in E} x_j \left(c_j - \sum_{i=1}^m \lambda_i a_{ij} \right) + \sum_{i=1}^m \lambda_i b_i,$$

где E — множество индексов j , для которых $\bar{x}_j = 0$. Поэтому

$$\begin{aligned} F_{\Lambda}(\bar{X}) - F_{\Lambda}(X) &= \sum_{j \in E} (\bar{x}_j - x_j) \left(c_j - \sum_{i=1}^m a_{ij} \lambda_i \right) = \\ &= - \sum_{j \in E} x_j \left(c_j - \sum_{i=1}^m a_{ij} \lambda_i \right). \end{aligned}$$

По условию $x_j \geq 0$ и множители λ_i удовлетворяют неравенствам (7.20).

Следовательно, $F_{\Lambda}(\bar{X}) - F_{\Lambda}(X) \geq 0$ для любых векторов с неотрицательными составляющими.

Достаточность. Предположим, что план \bar{X} задачи (5.1) — (5.3) удовлетворяет условию

$$F_{\Lambda}(\bar{X}) = \max_{X \geq 0} F_{\Lambda}(X) \quad (7.22)$$

при некотором векторе $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$.

Если при некотором j

$$c_j - \sum_{i=1}^m a_{ij} \lambda_i > 0,$$

то, увеличивая безгранично компоненту x_j вектора X , мы получим

$$\sup_{X \geq 0} F_{\Lambda}(X) = \infty,$$

что противоречит условию (7.22). Следовательно,

$$c_j - \sum_{i=1}^m a_{ij} \lambda_i \leq 0 \quad \text{для } j = 1, 2, \dots, n. \quad (7.23)$$

Если при некотором j , для которого $\bar{x}_j > 0$,

$$c_j - \sum_{i=1}^m a_{ij} \lambda_i < 0,$$

то, полагая

$$X^* = (\bar{x}_1, \dots, \bar{x}_{j-1}, 0, \bar{x}_{j+1}, \dots, \bar{x}_n),$$

имеем $F_{\Lambda}(X^*) > F_{\Lambda}(\bar{X})$, что противоречит (7.22).

Следовательно,

$$c_j - \sum_{i=1}^m a_{ij} \lambda_i = 0 \quad \text{при } \bar{x}_j > 0. \quad (7.24)$$

В соответствии с теоремой 7.4 соотношения (7.23) и (7.24) указывают на оптимальность плана \bar{X} . При этом вектор Λ , участвующий в построении функции F_{Λ} , является разрешающим вектором задачи. Теорема доказана.

Доказанное утверждение дает основание называть компоненты вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, участвующего в образовании функции $F_{\Lambda}(X)$ для задачи (5.1)–(5.3), множителями Лагранжа этой задачи. В процессе доказательства теоремы 7.5 было, в частности, установлено, что совокупность разрешающих векторов задачи линейного программирования совпадает с системой векторов, составленных из множителей Лагранжа данной задачи. Итак, разрешающие множители и множители Лагранжа

задачи линейного программирования — понятия эквивалентные.

В формулировке теоремы 7.5 вектор \bar{X} предполагался планом рассматриваемой задачи. Поэтому эта теорема еще не освобождает нас полностью от необходимости учитывать условия (5.2), связывающие переменные задачи (5.1) — (5.3).

Для того чтобы освободиться от ограничений (5.2), необходимо рассмотреть задачу об отыскании седловой точки для функции Лагранжа.

Дадим соответствующее определение. Пусть $R(X, Y)$ — функция, зависящая от вектора X , принадлежащего множеству T_X , и вектора Y , изменяющегося в пределах множества T_Y .

Точку $(X_0, Y_0) \in T_X \times T_Y^*$ назовем *седловой точкой* функции $R(X, Y)$ при условии $(X, Y) \in T_X \times T_Y$, если соотношения

$$R(X, Y_0) \leq R(X_0, Y_0) \leq R(X_0, Y) \quad (7.25)$$

имеют место для всех точек $(X, Y) \in T_X \times T_Y$. Неравенства (7.25) означают, что наибольшее значение функции $R(X, Y_0)$ на множестве T_X достигается в точке X_0 , а наименьшее значение функции $R(X_0, Y)$ на T_Y — в точке Y_0 .

Обратимся к задаче линейного программирования (6.20) — (6.22). Ограничения (6.20) задачи составлены из равенств и неравенств. Условие неотрицательности наложено лишь на часть переменных задачи. Пусть по-прежнему

$$\begin{aligned} F_{\Lambda}(X) = F(X, \Lambda) &= \sum_{j=1}^n c_j x_j + \sum_{i=1}^m \lambda_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) = \\ &= \sum_{j=1}^n c_j x_j + \sum_{i=1}^m \lambda_i b_i - \sum_{i=1}^m \sum_{j=1}^n \lambda_i x_j a_{ij}. \end{aligned}$$

Теорема 7.6. Векторы $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ и $\Lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)$ являются соответственно решением задачи (6.20) — (6.22) и ее разрешающим вектором в том и только в том случае, если (X^*, Λ^*) — седловая точка

*) По определению $W = (u, v) \in T_X \times T_Y$, если $u \in T_X$, $v \in T_Y$.

функции $F(X, \Lambda)$ при условиях

$$\left. \begin{aligned} x_j &\geq 0, & j &= 1, 2, \dots, n_1, \\ \lambda_i &\geq 0, & i &= 1, 2, \dots, m_1. \end{aligned} \right\} \quad (7.26)$$

Доказательство этой теоремы основывается на соображениях, близких к тем, которые были использованы при установлении предыдущей теоремы.

Согласно теореме 7.6 пара взаимосопряженных задач (6.20)—(6.22) и (6.23)—(6.25) эквивалентна задаче об отыскании седловой точки функции Лагранжа $F(X, \Lambda)$ при условиях (7.26).

Теорема 7.6, сформулированная для задач линейного программирования, может быть перенесена на широкий класс задач нелинейного программирования.

Указанное обобщение служит теоретической основой для построения большинства численных методов нелинейного программирования.

7.5. В заключение параграфа укажем на возможность еще одной интерпретации разрешающих множителей задачи линейного программирования. Ограничимся рассмотрением задачи (5.1)—(5.3), записанной в канонической форме.

Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — разрешающий вектор задачи (5.1)—(5.3), или, что то же самое, решение двойственной задачи (5.4)—(5.5). Мы покажем сейчас, что компоненты λ_i вектора Λ могут интерпретироваться как оценки влияния различных условий системы (5.2) на величину максимума задачи (5.1)—(5.3). Сформулируем это утверждение точнее.

Теорема 7.7. Пусть задача (5.1)—(5.3) невырождена и $M(b_1, b_2, \dots, b_m)$ обозначает максимум ее линейной формы при условиях (5.2), (5.3). В таком случае

$$\lambda_i = \frac{\partial M(b_1, b_2, \dots, b_m)}{\partial b_i}, \quad i = 1, 2, \dots, m. \quad (7.27)$$

Доказательство. Обозначим задачу (5.1)—(5.3) через (A_B) . Пусть $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ — опорное решение задачи (A_B) . Без ограничения общности можно считать, что отличными от нуля являются первые m

компонент вектора X^* . Тогда векторы условий A_j , $j = 1, 2, \dots, m$, линейно независимы и любой m -мерный вектор $B' = (b'_1, b'_2, \dots, b'_m)$ представим в виде их линейной комбинации

$$B' = \sum_{i=1}^m x'_i A_i. \quad (7.28)$$

Если через $\|e_{ij}\|_m$ обозначить матрицу, обратную невырожденной матрице (A_1, A_2, \dots, A_m) , то из системы (7.28) получаем

$$x'_i = \sum_{j=1}^m e_{ij} b'_j, \quad i = 1, 2, \dots, m. \quad (7.29)$$

Положим

$$e = \max_{1 \leq i \leq m} \sum_{j=1}^m |e_{ij}|, \quad x = \min_{1 \leq i \leq m} x_i^*. \quad (7.30)$$

Рассмотрим задачу линейного программирования $(A_{B'})$, образующуюся из задачи (A_B) заменой вектора ограничений B на B' . Проверим, что при

$$\max_{1 \leq i \leq m} |b_i - b'_i| \leq \frac{x}{e} \quad (7.31)$$

n -мерный вектор $X' = (x'_1, x'_2, \dots, x'_m, 0, 0, \dots, 0)$, где компоненты x'_i определяются формулой (7.29), является решением задачи $(A_{B'})$. Покажем вначале, что X' — план задачи $(A_{B'})$. В соответствии с соотношениями (7.29)

$$x'_i - x_i^* = \sum_{j=1}^m e_{ij} (b'_j - b_j), \quad i = 1, 2, \dots, m,$$

откуда, учитывая обозначения (7.30) и условие (7.31), имеем

$$|x'_i - x_i^*| \leq e \max |b'_j - b_j| \leq x$$

и, следовательно,

$$x'_i \geq x_i^* - x \geq 0.$$

Итак, вектор X' удовлетворяет условиям (5.3) задачи.

($A_{B'}$). Что касается условий (5.2), в которых вектор B заменен на B' , то они удовлетворяются вектором X' в соответствии с определением величин x'_1, x'_2, \dots, x'_m (см. соотношения (7.28)). Таким образом, X' является планом задачи ($A_{B'}$).

Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — оптимальный план задачи (5.4) — (5.5) (разрешающий вектор задачи (A_B)). Согласно критерию оптимальности плана задачи (A_B) (см. теорему 7.4) имеют место соотношения

$$\left. \begin{aligned} \sum_{i=1}^m a_{ij} \lambda_i &= c_j, & j &= 1, 2, \dots, m, \\ \sum_{i=1}^m a_{ij} \lambda_i &\geq c_j, & j &= m+1, m+2, \dots, n. \end{aligned} \right\} \quad (7.32)$$

Соотношения (7.32) показывают, что вектор Λ является также разрешающим вектором задачи ($A_{B'}$), связанным условиями (7.16) с ее планом X' . Следовательно, вектор X' является решением задачи ($A_{B'}$).

Итак, при выполнении требований (7.31) вектор X' является решением задачи ($A_{B'}$), а вектор Λ (не зависящий от выбора вектора B') составляет решение задачи, двойственной по отношению к задаче ($A_{B'}$).

Согласно первой теореме двойственности

$$M(b'_1, b'_2, \dots, b'_m) = \sum_{i=1}^m b'_i \lambda_i, \quad (7.33)$$

где вектор B' ограничен лишь условиями (7.31). Учитывая положительность величины $x = \min_{1 \leq i \leq m} x_i^*$ (план X^* , по предположению, невырожденный), приходим к выводу, что формула (7.33) имеет место для любого вектора B' , расположенного в некоторой фиксированной окрестности вектора B . Искомые равенства (7.27) можно получить теперь непосредственным дифференцированием соотношения (7.33). Теорема доказана.

Итак, в невырожденном случае компоненты разрешающего вектора задачи (5.1) — (5.3) оказываются оценками влияния правых частей условий (5.2) на величину максимально достижимого значения линейной формы (5.1).

При доказательстве теоремы 7.7 мы не использовали допущения о невырожденности всех опорных планов задачи. Предполагалось, что этому условию удовлетворяет только один из ее оптимальных планов. Следует отметить, что последнее условие является существенным.

Нарушение этого условия обычно влечет за собой неединственность решения двойственной задачи. Поэтому формула (7.27), вообще говоря, перестает быть верной: функция $M(b_1, \dots, b_m)$ может и не иметь частных производных. Однако и в вырожденном случае приведенная здесь интерпретация разрешающих множителей сохраняется (правда, в несколько модифицированной форме *).

*) См., например, книгу [12] (§ 5, гл. 3).

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ ПЛАНА

Методы решения задач линейного программирования делятся на конечные и итеративные. Каждый из конечных методов позволяет провести полное исследование задачи за конечное число однотипных шагов (итераций). Итеративные методы линейного программирования представляют собой бесконечные сходящиеся процессы вычисления оптимального плана задачи. Каждый шаг итеративного метода, как правило, гораздо проще отдельной итерации конечного метода. Итеративные методы — приближенные методы, однако при наличии достаточного времени они позволяют получить решение задачи с любой заранее заданной точностью.

В гл. 4—8 изучаются теоретические и вычислительные аспекты конечных методов линейного программирования. Гл. 9 посвящена итеративным методам.

Настоящая глава посвящена теоретическим основам метода последовательного улучшения плана. Этот конечный метод встречается в практике линейного программирования чаще других. Основы метода были сформулированы Данцигом в 1947 г. В иностранной литературе по линейному программированию метод Данцига известен под названием *симплексного* метода. Такое название возникло из геометрического истолкования первых частных задач, к которым он был применен *), и не соответствует существу метода.

*) Ограничения вида $\sum_{j=1}^n x_j = 1, x_j \geq 0 \ (j = 1, 2, \dots, n)$ определяют в n -мерном пространстве симплекс. Подобные ограничения входили в условия одной из первых задач линейного программирования, для которой Данциг разработал вычислительный метод.

Идея метода содержит три существенных момента. Во-первых, указывается способ вычисления опорного плана. Во-вторых, устанавливается признак, который позволяет проверить, является ли выбранный опорный план оптимальным. В-третьих, приводится способ, позволяющий по выбранному неоптимальному плану построить другой опорный план, более близкий к оптимальному. Доказывается, что таким путем можно через конечное число шагов получить оптимальный план — решение задачи линейного программирования. Таким образом, метод заключается в последовательном улучшении плана, что и целесообразно отразить в его названии.

Следует заметить, что алгоритмы метода позволяют также в процессе вычислений установить, является ли задача линейного программирования разрешимой. Это значит, что в ходе расчетов можно определить, не оказываются ли условия задачи противоречивыми и обеспечивают ли они ограниченность ее линейной формы.

§ 1. ПРИЗНАК ОПТИМАЛЬНОСТИ

1.1. В гл. 3 был получен критерий оптимальности плана — необходимое и достаточное условие того, что исследуемый план является решением задачи линейного программирования. В методе последовательного улучшения плана приходится оперировать не с произвольными планами, а только с опорными планами.

Ниже будет сформулирован признак оптимальности опорного плана. Признак оптимальности может быть выражен через те или иные параметры, характеризующие задачу. В соответствии с этим будем говорить о двух разных формах признака оптимальности. Каждая из форм признака будет далее положена в основу одного из алгоритмов решения задачи линейного программирования, реализующих метод последовательного улучшения плана. Термин «признак» (в отличие от термина «критерий») оптимальности принят для того, чтобы подчеркнуть, что здесь речь идет не о произвольном плане, а об опорном и что приведенное условие является, вообще говоря, только достаточным.

Запишем задачу линейного программирования в канонической форме. Требуется вычислить максимум линейной формы

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

при условиях

$$\sum_{j=1}^n A_j x_j = B, \quad (1.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (1.3)$$

Более компактно задача записывается в следующем виде. Требуется вычислить неотрицательный вектор (вектор-строку) $X \geq 0$, обращающий в максимум $L(X) = CX^T$ при условиях $AX^T = B$. Здесь $A = (A_1, \dots, A_n) = \|a_{ij}\|_{mn}$ — матрица условий, A_j и B — соответственно векторы условий и вектор ограничений данной задачи:

$$A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T, \quad j = 1, 2, \dots, n,$$

$$B = (b_1, b_2, \dots, b_m)^T,$$

$C = (c_1, c_2, \dots, c_n)$ — заданный вектор коэффициентов линейной формы (1.1). Будем предполагать, что ранг матрицы A равен m .

Напомним определения опорного плана и его базиса — основных понятий настоящей главы. План $X = (x_1, x_2, \dots, x_n)$ задачи линейного программирования (записанной в канонической форме (1.1) — (1.3)) называется *опорным*, если система векторов условий A_j , соответствующая его положительным компонентам ($x_j > 0$), линейно независима.

Базисом опорного плана мы называли систему m линейно независимых векторов условий, которая включает все векторы A_j , отвечающие положительным составляющим опорного плана.

Компоненты опорного плана, отвечающие векторам его базиса, будем называть *базисными*, а остальные составляющие — *внебазисными* переменными.

Допустим, что нам известен некоторый опорный план X с базисом $A_{s_1}, A_{s_2}, \dots, A_{s_i}, \dots, A_{s_m}$. В дальнейшем будет удобно характеризовать вектор базиса, помимо его

номера s_i , позицией i , которую он занимает в рассматриваемом базисе. (Ясно, что при любом базисе $i = 1, 2, \dots, m$.) Зафиксируем произвольный вектор условий A_j и выпишем его разложение по векторам базиса A_{s_1}, \dots, A_{s_m} :

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

Здесь x_{ij} — составляющая разложения векторов A_j по векторам базиса при векторе A_{s_i} , расположенном в i -й позиции базиса.

Введем множество индексов I_X вектора базиса опорного плана X (I_X — множество номеров базисных переменных). Очевидно,

$$B = \sum_{j=1}^n x_j A_j = \sum_{j \in I_X} x_j A_j$$

(внебазисные переменные равны нулю).

Разложение вектора ограничений B по векторам базиса можно также записать в виде

$$B = \sum_{i=1}^m A_{s_i} x_{i0}.$$

$x_{i0} = x_{s_i}$ — базисная переменная, отвечающая вектору A_{s_i} , расположенному в i -й позиции базиса. Обозначив вектор ограничений B через A_0 , получим общую формулу

$$A_j = \sum_{i=1}^m A_{s_i} x_{ij}, \quad j = 0, 1, 2, \dots, m, \quad (1.4)$$

определяющую составляющие разложения всех векторов условий и вектора ограничений по векторам базиса.

Составим наборы параметров z_j и Δ_j ($j = 1, 2, \dots, n$) по следующим формулам:

$$z_j = \sum_{i=1}^m c_{s_i} x_{ij}, \quad j = 1, 2, \dots, n, \quad (1.5)$$

$$\Delta_j = z_j - c_j, \quad j = 1, 2, \dots, n. \quad (1.6)$$

Величины z_j и Δ_j определяются опорным планом X , соответствующим базису A_{s_1}, \dots, A_{s_m} . Чтобы подчеркнуть

это, целесообразно было бы обозначать эти параметры соответственно через $z_j^{(X)}$ и $\Delta_j^{(X)}$. Однако, не желая усложнять записи, будем всюду, где это не вызовет недоумений, опускать индекс X при z_j и Δ_j .

В методе последовательного улучшения плана параметры Δ_j играют очень важную роль. Знаки параметров Δ_j позволяют определить, является ли выбранный опорный план оптимальным.

Имеет место следующее предложение.

Признак оптимальности. *Опорный план X^* является решением задачи (1.1) — (1.3), если $\Delta_j \geq 0$ для $j = 1, 2, \dots, n$.*

Доказательство. Рассмотрим произвольный план $X = (x_1, \dots, x_n) \neq X^*$. Составляющие вектора X , как и компоненты всякого плана, должны удовлетворять условиям задачи, т. е.

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

Разложим вектор A_j согласно формуле (1.4) по векторам базиса опорного плана X^* . Получим

$$B = \sum_{j=1}^n x_j \left(\sum_{i=1}^m x_{ij} A_{s_i} \right),$$

или, меняя порядок суммирования,

$$B = \sum_{i=1}^m \left(\sum_{j=1}^n x_j x_{ij} \right) A_{s_i}. \quad (1.7)$$

С другой стороны, поскольку X^* является опорным планом задачи (1.1) — (1.3),

$$B = \sum_{i=1}^m x_{i0}^* A_{s_i}. \quad (1.8)$$

Векторы $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ составляют линейно независимую систему. Следовательно, вектор ограничений B может быть единственным образом представлен в виде их

линейной комбинации. Сравнивая (1.7) и (1.8), заключаем, что

$$\sum_{j=1}^n x_j x_{ij} = x_{i0}^*. \quad (1.9)$$

Справедливость сформулированного утверждения вытекает теперь из следующей цепочки равенств и неравенств:

$$\begin{aligned} L(X) &= \sum_{(1) j=1}^n c_j x_j \leq \sum_{(2) j=1}^n z_j x_j = \sum_{(3) j=1}^n \left(\sum_{i=1}^m c_{si} x_{ij} \right) x_j = \\ &= \sum_{(4) i=1}^m \left(\sum_{j=1}^n x_j x_{ij} \right) c_{si} = \sum_{(5) i=1}^m c_{si} x_{i0}^* = L(X^*). \end{aligned} \quad (6)$$

Здесь равенства (1) и (6) определяют значение линейной формы L для планов X и X^* соответственно. Неравенство (2) следует из условия $\Delta_j = z_j - c_j \geq 0$. Равенства (3) и (5) вытекают из формул (1.5) и (1.9) соответственно, а равенство (4) получено изменением порядка суммирования.

Таким образом, при $\Delta_j \geq 0$ для всех j опорный план X^* определяет максимальное значение линейной формы $L(X)$ и является решением задачи линейного программирования. Подчеркнем, что условие $\Delta_j \geq 0$ для всех j является, вообще говоря, только достаточным условием оптимальности опорного плана.

Как уже отмечалось, Δ_j можно выразить через различные параметры, характеризующие задачу линейного программирования. В зависимости от того, как вычисляется z_j в выражении (1.6) для Δ_j , мы имеем ту или иную форму признака оптимальности. Будем говорить, что признак оптимальности записывается в первой форме, если z_j вычисляются из соотношений (1.5). Таким образом, первая форма признака оптимальности опорного плана формулируется следующим образом. Опорный план X^* является решением задачи (1.1) — (1.3), если

$$\sum_{i=1}^m c_{si} x_{ij} \geq c_j$$

для всех j ($j = 1, 2, \dots, n$).

1.2. Приведем еще одну форму признака оптимальности опорного плана.

Пусть по-прежнему векторы условий $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ образуют базис опорного плана X^* . Определим вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ из системы уравнений

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j, \quad j \in I_X. \quad (1.10)$$

Вторая форма признака оптимальности опорного плана может быть сформулирована следующим образом. Опорный план X^* значения (1.1) — (1.3) оптимален, если

$$\sum_{i=1}^m a_{ij} \lambda_i \geq c_j \quad (1.11)$$

для $j = 1, 2, \dots, n$. Справедливость этого достаточного условия оптимальности опорного плана легко доказывается, исходя из критерия оптимальности плана, установленного в § 7 гл. 3. Обоснование приведенного признака вытекает также из следующей цепочки равенств:

$$\begin{aligned} z_j &= \sum_{(1) i=1}^m c_{s_i} x_{ij} = \sum_{(2) i=1}^m \left(\sum_{\mu=1}^m a_{\mu s_i} \lambda_{\mu} \right) x_{ij} = \sum_{(3) \mu=1}^m \lambda_{\mu} \sum_{i=1}^m a_{\mu s_i} x_{ij} = \\ &= \sum_{(4) \mu=1}^m \lambda_{\mu} a_{\mu j}. \end{aligned}$$

Здесь равенство (1) совпадает с определением (1.5) величин z_j . Равенство (2) вытекает из (1.10). Равенство (3) получено изменением порядка суммирования. Равенство (4) следует из разложения (1.4), если его записать для компонент векторов условий.

Теперь видно, что соотношение

$$\sum_{i=1}^m a_{ij} \lambda_i \geq c_j$$

для всех j эквивалентно условию

$$\Delta_j \geq 0 \quad \text{для } j = 1, 2, \dots, n.$$

Таким образом, доказана не только справедливость второй формы признака оптимальности, но и эквивалентность обеих форм.

¹ Как уже отмечалось, каждой форме признака оптимальности соответствует своя вычислительная схема решения задач линейного программирования.

§ 2. ОБЩАЯ СХЕМА МЕТОДА

2.1. Приведем общую схему решения задачи линейного программирования методом последовательного улучшения плана применительно к невырожденному случаю. Напомним, что опорный план задачи (1.1) — (1.3) называется невырожденным, если число его положительных компонент в точности равно m . Задача линейного программирования называется невырожденной, если все ее опорные планы не вырождены.

Пусть известен некоторый опорный план X задачи с базисом $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Анализ плана начинается с проверки его на оптимальность. Проверка может производиться в соответствии с первой или второй формой признака оптимальности. И в том и в другом случае необходимо вычислить значения $\Delta_j (j=1, 2, \dots, n)$. В первом случае Δ_j вычисляется по формуле

$$\Delta_j = z_j - c_j = \sum_{i=1}^m c_{s_i} x_{ij} - c_j, \quad (2.1)$$

где x_{ij} определяется разложением (1.4).

При использовании второй формы признака Δ_j определяется соотношением

$$\Delta_j = z_j - c_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j, \quad (2.2)$$

где $\Lambda = (\lambda_1, \dots, \lambda_m)$ — решение системы (1.10).

При проверке опорного плана на оптимальность может встретиться один из следующих трех случаев:

1°. $\Delta_j \geq 0$ для $j \notin I_X$.

(При $j \in I_X$ $z_j = c_j$). Поэтому в случае 1° величины $\Delta_j \geq 0$ для всех j от 1 до n).

2°. $\Delta_j < 0$ для некоторого j , и все соответствующие этому индексу величины $x_{ij} \leq 0$ ($i=1, 2, \dots, m$).

3°. $\Delta_j < 0$ для некоторых индексов j , и для каждого такого j по крайней мере одно из чисел x_{ij} положительно.

В случае 1°, как это следует из признака оптимальности опорного плана, план X является решением задачи линейного программирования.

Мы сейчас покажем, что в случае 2° задача линейного программирования неразрешима, а в случае 3° можно указать способ перехода от опорного плана X к новому опорному плану, более близкому к решению задачи.

2.2. Проследим, как изменяется линейная форма (1.1) в результате перехода к некоторому новому плану $X(\theta)$, который образуется из плана X в соответствии со следующим правилом:

а) j -я компонента плана $X(\theta)$ полагается равной некоторому положительному числу θ (j — фиксированный индекс внебазисной переменной плана X): $x_j(\theta) = \theta$;

б) остальные внебазисные переменные плана X остаются равными нулю ($x_s(\theta) = x_s = 0$ при $s \notin I_X$, $s \neq j$);

в) базисные составляющие $X(\theta)$ выбираются так, чтобы вектор $X(\theta) = (x_1(\theta), \dots, x_n(\theta))$ определял план задачи, т. е. чтобы числа $x_j(\theta)$ ($j = 1, 2, \dots, n$) удовлетворяли условиям (1.2) — (1.3).

Будем в дальнейшем указанный переход от плана X к плану $X(\theta)$ называть *элементарным преобразованием*, связанным с вектором A_j . Выпишем соотношения, определяющие элементарное преобразование. В силу определения опорного плана X и предположений о структуре плана $X(\theta)$ имеем

$$\sum_{i=1}^m x_{i0} A_{s_i} = B, \quad (2.3)$$

$$\sum_{i=1}^m x_{i0}(\theta) A_{s_i} + \theta A_j = B. \quad (2.4)$$

Согласно формуле (1.4) разложение произвольного вектора условий по векторам базиса записывается в виде

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}. \quad (1.4)$$

Из формул (1.4) и (2.3) получаем

$$\sum_{i=1}^m (x_{i0} - \theta x_{ij}) A_{s_i} + \theta A_j = B. \quad (2.5)$$

Сравнивая полученное соотношение с системой (2.4) и учитывая единственность ее решения (векторы A_{s_1}, \dots, A_{s_m} , составляющие базис опорного плана X , линейно независимы), приходим к равенствам, определяющим элементарное преобразование,

$$x_{i0}(\theta) = x_{i0} - \theta x_{ij}. \quad (2.6)$$

Напомним, что в соответствии с определением элементарного преобразования, связанного с вектором A_j ,

$$x_t(\theta) = x_t = 0 \text{ при } t \notin I_X, \quad t \neq j; \quad x_j(\theta) = \theta.$$

Таким образом, элементарное преобразование полностью определено.

Все x_{i0} — базисные компоненты плана X — положительны. Следовательно, до тех пор, пока θ настолько мало, что все величины $x_{i0}(\theta)$ из (2.6) неотрицательны, вектор $X(\theta)$ удовлетворяет условиям (1.2), (1.3) и является планом задачи линейного программирования. По составляющим плана $X(\theta)$ нетрудно вычислить соответствующее ему значение линейной формы (1.1):

$$L[X(\theta)] = \sum_{i=1}^m c_{s_i} x_{i0}(\theta) + c_j \theta = \sum_{i=1}^m c_{s_i} x_{i0} - \theta \left(\sum_{i=1}^m c_{s_i} x_{ij} - c_j \right).$$

Используя соотношения (1.5) и (1.6), получаем

$$L[X(\theta)] = L(X) - \theta \Delta_j. \quad (2.7)$$

Равенство (2.7) показывает, что влияние элементарного преобразования плана X на величину линейной формы определяется знаком Δ_j . Если $\Delta_j < 0$, значение линейной формы увеличивается, если $\Delta_j > 0$ — уменьшается. Если $\Delta_j = 0$, линейная форма сохраняет свое прежнее значение.

2.3. Введем некоторые термины, удобные для последующего изложения. Параметры Δ_j , знаки которых определяют направление изменения линейной формы при элементарном преобразовании, связанном с вектором A_j , естественно называть *оценками векторов условий A_j относительно данного базиса*. (Поскольку в формировании Δ_j участвуют и коэффициенты c_j линейной формы, более точным наименованием параметров Δ_j было бы «оценки

расширенных векторов условий \bar{A}_j относительно рассматриваемого базиса.) Целесообразность введенного термина определяется следующими соображениями.

Элементарные преобразования, связанные с векторами, отвечающими $\Delta_j < 0$ и $\Delta_j > 0$, соответственно увеличивают и уменьшают значение линейной формы. При этом величина изменения линейной формы при $\theta = 1$ в точности совпадает со значением Δ_j . Таким образом, параметры Δ_j оценивают изменение линейной формы, которое произойдет, если ввести вектор A_j в рассматриваемый базис. Этот факт и предлагается отразить в названии параметров Δ_j . Из приведенных рассуждений следует, что определение $\Delta_j = c_j - z_j$ больше соответствует понятию оценки вектора базиса, чем принятое здесь определение $\Delta_j = z_j - c_j$. Тем не менее, чтобы избежать возможных недоразумений, мы сохраняем здесь уже установившееся в литературе по линейному программированию понятие $\Delta_j = z_j - c_j$. Это позволит также сделать вычислительные схемы более однообразными.

При использовании второй формы признака оптимальности величины Δ_j вычисляются по формуле

$$\Delta_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j.$$

Фигурирующие здесь параметры λ_i будем в дальнейшем называть *оценками условий задачи относительно данного базиса*. Чтобы обосновать целесообразность введенного термина, вспомним определение двойственной задачи.

Задача минимизации линейной формы $\sum_{i=1}^m b_i y_i$ при условиях $\sum_{i=1}^m a_{ij} y_i \geq c_j$ двойственна по отношению к задаче (1.1) — (1.3).

Из второй формы признака оптимальности опорного плана следует, что параметры λ_i^* , отвечающие базису опорного решения X^* , удовлетворяют условиям двойственной задачи

$$\Delta_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j \geq 0.$$

С другой стороны, для оптимального опорного плана X^*

$$\begin{aligned} L(X^*) &= \sum_{j=1}^n c_j x_j^* = \sum_{\mu=1}^m c_{s_\mu} x_{\mu 0}^* = \sum_{\mu=1}^m \left(\sum_{i=1}^m \lambda_i^* a_{is_\mu} \right) x_{\mu 0}^* = \\ &= \sum_{i=1}^m \lambda_i^* \left(\sum_{\mu=1}^m a_{is_\mu} x_{\mu 0}^* \right) = \sum_{i=1}^m \lambda_i^* b_i. \end{aligned}$$

Отсюда вытекает (см. лемму 5.2 гл. 3), что вектор $\Lambda^* = (\lambda_1^*, \dots, \lambda_m^*)$ является решением двойственной задачи. В предыдущей главе было признано целесообразным называть составляющие оптимального плана двойственной задачи оценками условий исходной задачи (оценками влияния правых частей условий (1.2) на максимально достижимое значение линейной формы (1.1)). Выбор λ_i^* обеспечивает равенство нулю всех Δ_j^* , отвечающих векторам базиса решения. Параметры λ_i , соответствующие произвольному опорному плану, играют по отношению к базису этого плана ту же роль, что и λ_i^* по отношению к базису оптимального опорного плана. Как видно из определения λ_i , выбор этих параметров обеспечивает равенство нулю чисел Δ_j — оценок векторов условий базиса. Приведенные рассуждения служат основанием для того, чтобы называть параметры λ_i оценками условий задачи относительно выбранного базиса.

2.4. Формулы (2.6) и (2.7) позволяют проанализировать все случаи, возникающие в процессе проверки опорного плана на оптимальность.

В случае 1°, как мы уже видели, в соответствии с признаком оптимальности план X является решением задачи линейного программирования.

В случае 2° имеется индекс $j=k$, для которого $\Delta_k < 0$ и все соответствующие компоненты $x_{ik} \leq 0$ ($i=1, 2, \dots, m$). Формула (2.6) показывает ($j=k$), что в этом случае $X(\theta)$ является планом задачи (1.1) — (1.3) при любом $\theta > 0$ ($x_j(\theta) \geq 0$ при любом $\theta \geq 0$). А это означает, как видно из формулы (2.7), что линейная форма $L(X)$ не ограничена сверху на множестве планов задачи. Таким образом, в случае 2° задача линейного программирования неразрешима.

Чаще всего приходится сталкиваться с третьей возможностью. В этом случае $\Delta_j < 0$ для некоторых j , и для каждого такого j по крайней мере одно из чисел x_{ij} положительно. Покажем, что при этих условиях можно с помощью элементарного преобразования, связанного с вектором A_k ($\Delta_k < 0$), перейти от опорного плана X к новому опорному плану X' и увеличить при этом значение линейной формы задачи.

Проведем элементарное преобразование плана X при значении θ , равном минимуму отношения x_{i0}/x_{ik} по всем i из множества $1, 2, \dots, m$, для которых $x_{ik} > 0$ (по условию такие x_{ik} имеются).

Итак, пусть

$$\theta_0 = \min_{\substack{i \\ x_{ik} > 0}} \frac{x_{i0}}{x_{ik}}.$$

По условию $x_{i0} > 0$ для $i = 1, 2, \dots, m$. Поэтому θ_0 положительно. Здесь мы существенно используем невырожденность опорного плана X .

Положим $X' = X(\theta_0)$. По построению $X(\theta_0)$ является планом задачи. Более того, мы сейчас установим, что X' является опорным планом задачи. Пусть минимум отношения x_{i0}/x_{ik} , определяющий число θ_0 , достигается при $i = r$. Очевидно, $x'_{s_r} = x_{s_r}(\theta_0) = 0$. Мы будем иногда говорить, что θ_0 достигается на r -й позиции базиса, или, что то же самое, на векторе условий A_{s_r} .

В силу (2.4) при выбранном $\theta = \theta_0$ вектор ограничений B оказывается линейной комбинацией векторов $A_{s_1}, A_{s_2}, \dots, A_{s_{r-1}}, A_{s_{r+1}}, \dots, A_{s_m}, A_k$ с неотрицательными коэффициентами. Нетрудно убедиться в том, что эта система векторов линейно независима. Справедливость этого утверждения вытекает из следующей простой теоремы, которой мы будем неоднократно пользоваться и в дальнейшем.

Теорема 2.1. Если P_1, P_2, \dots, P_m — линейно независимая система векторов и вектор $Q = \sum_{i=1}^m \zeta_i P_i$, причем $\zeta_r \neq 0$, то система $P_1, P_2, \dots, P_{r-1}, Q, P_{r+1}, \dots, P_m$ также линейно независима.

Доказательство. Пусть сформулированное утверждение неверно. Тогда можно указать числа $\alpha_1, \alpha_2, \dots, \alpha_{r-1}, \alpha_{r+1}, \dots, \alpha_m, \alpha$, среди которых имеются и ненулевые, такие, что

$$\sum_{\substack{i=1 \\ i \neq r}}^m \alpha_i P_i + \alpha Q = 0.$$

Подставляя в эту формулу разложение Q по векторам системы P_1, P_2, \dots, P_m , получаем

$$\sum_{\substack{i=1 \\ i \neq r}}^m (\alpha_i + \alpha \zeta_i) P_i + \alpha \zeta_r P_r = 0.$$

Система векторов $P_1, \dots, P_r, \dots, P_m$ по условию линейно независима, а $\zeta_r \neq 0$. Это значит, что последнее соотношение выполняется лишь при $\alpha = \alpha_1 = \dots = \alpha_{r-1} = \alpha_{r+1} = \dots = \alpha_m = 0$. Мы пришли к противоречию. Теорема доказана.

Из доказанной теоремы следует линейная независимость системы $A_{s_1}, \dots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \dots, A_{s_m}$ (здесь P_1, \dots, P_m — векторы базиса, $Q = A_k$, $\zeta_r = x_{rk} > 0$). Это значит, что план $X' = X(\theta_0)$ действительно является опорным планом задачи и его базис состоит из векторов $A_{s_1}, \dots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \dots, A_{s_m}$.

Таким образом, базис нового опорного плана образуется из базиса предыдущего опорного плана заменой вектора A_{s_r} , расположенного в r -й позиции базиса (на которой достигается θ_0), на вектор A_k с отрицательной относительной оценкой. Вектор A_k занимает в новом базисе позицию r . В соответствии с этим

$$x'_{i0} = \begin{cases} x_{i0}(\theta_0), & \text{если } i \neq r, \\ \theta_0 & \text{при } i = r. \end{cases}$$

Используя формулу (2.6), получаем выражения для базисных переменных нового опорного плана:

$$x'_i = \begin{cases} x_{i0} - \theta_0 x_{ik}, & i \neq r, \\ \theta_0, & i = r. \end{cases} \quad (2.8)$$

Заметим, что в невырожденной задаче величина θ_0 может достигаться только при одном значении i , т. е. только на одной позиции базиса. В противном случае опорный план X' оказался бы вырожденным, поскольку он имел бы менее чем m положительных компонент.

В соответствии с соотношением (2.7)

$$L(X') = L(X) - \theta_0 \Delta_k.$$

По условию $\Delta_k < 0$. Кроме того, как мы видели, θ_0 — положительное число. Поэтому

$$L(X') > L(X).$$

Таким образом, в случае 3° можно перейти от исходного опорного плана X к новому опорному плану X' , более близкому к решению задачи. Переход к новому плану приводит к увеличению линейной формы на $-\theta_0 \Delta_k$. Последовательные переходы от одного опорного плана к следующему производятся до тех пор, пока будет получено решение задачи, либо будет установлена ее неразрешимость.

Каждый переход от одного опорного плана к следующему составляет *итерацию (шаг)* метода последовательного улучшения плана. Количество итераций, в результате которых мы приходим к решению невырожденной задачи или доказываем ее неразрешимость, конечно. Действительно, каждому опорному плану невырожденной задачи соответствует своя система из m линейно независимых векторов условий, составляющих его базис. Общее количество векторов условий равно n . Поэтому существует не более C_n^m различных базисов. Таким образом, и число различных опорных планов задачи конечно. Для каждого опорного плана однозначно определяется величина линейной формы. В невырожденном случае каждый следующий шаг увеличивает значение линейной формы. Поэтому не может быть возвращения к ранее пройденному базису. Это значит, что через конечное число итераций (шагов) метода последовательного улучшения плана будет получено решение (случай 1°) или установлена неограниченность линейной формы на множестве планов задачи (случай 2°).

Элементарное преобразование опорного плана носит достаточно частный характер. Тем не менее, как мы видели, с помощью элементарных преобразований всегда может быть осуществлено построение оптимального плана исследуемой задачи. Процесс решения задачи линейного программирования методом последовательного улучшения плана складывается из ряда таких преобразований.

2.5. Повторим кратко порядок операций, связанных с отдельным шагом метода последовательного улучшения плана.

Каждая итерация метода содержит два этапа. Первый этап состоит в проверке исследуемого опорного плана на оптимальность. Первый этап приводит к одной из трех возможностей (случаи 1°, 2°, 3°). Если имеет место случай 1° или 2°, процесс решения заканчивается. В случае 1° исследуемый план оптимален. В случае 2° установлена неразрешимость задачи. Если имеет место случай 3°, переходим ко второму этапу итерации. Второй этап заключается в определении элементарного преобразования, приводящего к новому опорному плану с большим значением линейной формы. На втором этапе выбирается вектор A_k с отрицательной оценкой ($\Delta_k < 0$) и определяется позиция r , которую он должен занять в базисе. Вектор A_{s_r} исключается из базиса. Новый базис образуется, таким образом, из векторов старого базиса заменой вектора условий A_{s_r} на вектор A_k . На втором этапе вычисляются все параметры, необходимые для получения и проверки на оптимальность нового опорного плана. Проверка полученного опорного плана на оптимальность производится на первом этапе следующей итерации.

Все рассуждения настоящего параграфа проводились для задачи линейного программирования, в которой требовалось обратить в максимум линейную форму (1.1) при условиях (1.2), (1.3). Нет необходимости проводить специальные рассуждения для задачи минимизации линейной формы. Всякая задача, в которой требуется определить минимум линейной формы $L(X) = \sum_{j=1}^n c_j x_j$ при

некоторых условиях, сводится к задаче максимизации формы $\bar{L}(X) = - \sum_{j=1}^n c_j x_j$ при тех же ограничениях.

§ 3. ГЕОМЕТРИЧЕСКИЕ ИНТЕРПРЕТАЦИИ МЕТОДА

3.1. Все элементы метода последовательного улучшения плана находят простое геометрическое истолкование. В гл. 3 были приведены две геометрические интерпретации задачи линейного программирования. Читателю полезно будет вернуться к ним и связать изложенное в настоящей главе алгебраическое описание метода с соответствующими геометрическими образами. Геометрические описания метода будут проводиться в терминах многомерных пространств. При желании читатель может ограничиться двух- или трехмерным случаем, где все сказанное ниже приобретает геометрическую наглядность.

Различные геометрические интерпретации задачи приводят к разным геометрическим истолкованиям метода. В соответствии с первой геометрической интерпретацией задачи линейного программирования условия (1.2)—(1.3) высекают в пространстве переменных x_1, \dots, x_n выпуклый многогранник (или неограниченное выпуклое многогранное множество), размерность которого не превышает $n - m$. Это многогранное множество содержится в общей части гиперплоскостей, определяемых условиями (1.2) задачи.

Опорному плану X соответствует вершина многогранного множества. Будем ее обозначать, как и опорный план, буквой X . Вершина X образуется пересечением гиперплоскостей, отвечающих условиям (1.2), и гиперплоскостей, соответствующих нулевым компонентам опорного плана. В невырожденном случае все базисные переменные отличны от нуля. Опорный план содержит $n - m$ внебазисных компонент. Таким образом, в невырожденном случае в вершине X пересекаются ровно $m + (n - m) = n$ независимых гиперплоскостей.

Выберем одну из внебазисных переменных (x_j) и заменим условие $x_j = 0$ ограничением $x_j \geq 0$. Если все остальные внебазисные переменные сохраняют нулевое значение, то полученное таким образом геометрическое

место точек является лучом, направленным «внутри» многогранного множества (где $x_j > 0$). Будем говорить, что этот луч отвечает вектору условий A_j . Геометрическое место точек, в которых линейная форма $L = \sum_{j=1}^n c_j x_j$

сохраняет постоянное значение, представляет собой гиперплоскость. Будем называть ее гиперплоскостью линейной формы. Вектор $C = (c_1, c_2, \dots, c_n)$ определяет направление, в котором следует смещать гиперплоскость, чтобы увеличить значение L . Гиперплоскость линейной формы делит пространство x_1, x_2, \dots, x_n на два полупространства. Полупространство, в котором расположен вектор C , будем называть «верхним» полупространством, а противоположное — «нижним».

Элементарное преобразование опорного плана X , связанное с вектором A_j , в геометрических терминах означает перемещение из вершины X вдоль луча, отвечающего вектору A_j . Каждое ребро выпуклого многогранного множества, исходящее из вершины X , расположено на пересечении некоторых $n-1$ независимых граничных гиперплоскостей, содержащих точку X . Поэтому в невырожденном случае любому такому ребру соответствует луч, определяемый одним из векторов $A_j, j \in I_X$. С другой стороны, в рассматриваемом случае общая часть луча, отвечающего любому вектору $A_j, j \in I_X$, и многогранного множества содержит отрезок ($\theta_0 > 0$) и, следовательно, составляет ребро данного множества. Итак, при невырожденном плане X между векторами $A_j, j \in I_X$, и ребрами многогранного множества, исходящими из точки X , устанавливается взаимно однозначное соответствие. Напомним, что ребра выпуклого многогранного множества могут быть ограниченными и неограниченными. В первом случае ребро — отрезок, соединяющий две вершины множества. Во втором — луч, исходящий из некоторой вершины. Мы можем теперь говорить, что каждое семейство элементарных преобразований при $0 \leq \theta \leq \theta_0$ геометрически эквивалентно движению вдоль некоторого ребра многогранного множества. Если число $\theta_0 < \infty$, то соответствующее ребро ограниченное. Величина $\theta_0 = \infty$ отвечает неограниченному ребру.

При фиксированном значении θ мы переходим в точку $X(\theta)$, что приводит к изменению линейной формы на Δ_θ .

Сформулируем в геометрических терминах все элементы отдельной итерации метода последовательного улучшения плана.

Первый этап итерации — это исследование опорного плана X на оптимальность. Проведем через вершину X гиперплоскость линейной формы и отберем все ребра многогранного множества, исходящие из точки X и расположенные в «верхнем» полупространстве. Если таких ребер нет, т. е. все ребра, содержащие вершину X , расположены «под» гиперплоскостью линейной формы, то опорный план X является оптимальным планом (случай 1°). Геометрически ясно, что если все ребра, исходящие из вершины X многогранного множества, лежат «ниже» гиперплоскости линейной формы, то и все множество лежит «под» гиперплоскостью. Этот факт и устанавливает признак оптимальности.

Допустим, что в «верхнем» полупространстве имеются ребра многогранного множества. При этом могут встретиться две возможности. Если среди ребер в «верхнем» полупространстве имеется неограниченное ребро (луч), то линейная форма, очевидно, не ограничена сверху (случай 2°). В том случае, когда все ребра многогранного множества в верхнем полупространстве ограничены (отрезки), можно перейти ко второму этапу итерации, сдвинуться вдоль одного из этих ребер и увеличить при этом значение линейной формы (случай 3°). Перемещение гиперплоскости линейной формы параллельно самой себе производится вдоль ребра до тех пор, пока она не пересечет другой конец ребра — соседнюю вершину X' . В точке X' одна из базисных переменных старого базиса обращается в нуль. Вершина X' — образ нового опорного плана X' — является, таким образом, пересечением гиперплоскостей, образующих ребро, и гиперплоскости, отвечающей старой базисной переменной, которая обратилась в нуль. Ясно, что $L(X') > L(X)$.

Перемещая таким образом гиперплоскость $L(X) = \text{const}$ параллельно самой себе от одной вершины многогранника условий к соседней, мы увеличиваем от шага к шагу значение линейной формы. Конечное число вершин много-

гранника условий (многогранного множества) гарантирует достижение максимума линейной формы за конечное число шагов. Из геометрических соображений ясно также, что неразрешимость задачи (неограниченность линейной формы в области ее определения), если она имеет место, обнаруживается опять-таки через конечное число шагов.

3.2. Приведем теперь геометрическое истолкование метода последовательного улучшения плана, соответствующее второй геометрической интерпретации задачи линейного программирования.

Напомним, что расширенные векторы условий \bar{A}_j порождают в $(m+1)$ -мерном пространстве точек $U=(u_1, u_2, u_3, \dots, u_{m+1})$ выпуклый многогранный конус K . Векторам условий A_j и вектору ограничений B в пространстве U соответствуют точки, у которых $u_{m+1}=0$. Прямая Q проходит через точку, определяемую вектором ограничений B , параллельно координатной оси Ou_{m+1} . Если прямая Q не пересекает конус K , множество планов задачи пусто — условия задачи противоречивы.

Нас будут интересовать только случаи, когда множество планов задачи не пусто, т. е. когда прямая Q и конус K имеют общую часть. Часть прямой Q , принадлежащая конусу K , является образом области определения линейной формы в $(m+1)$ -мерном пространстве точек U . Самая «верхняя» точка общей части K и Q (точка отрезка, отвечающая максимальному значению $(m+1)$ -й координаты) определяет максимум линейной формы, а самая «нижняя» точка отрезка — минимум L . (Естественно, речь идет о случае, когда такие точки существуют.)

Может оказаться, что общей частью прямой Q и конуса K является луч, направленный «вверх» или «вниз». В этом случае линейная форма не ограничена сверху или снизу на множестве планов задачи. Если вся прямая Q принадлежит конусу K , то линейная форма не ограничена с обеих сторон в области своего определения.

Укажем геометрические построения, к которым приводит реализация каждой итерации метода последова-

тельного улучшения плана. Пусть векторы условий A_1, A_2, \dots, A_m образуют базис некоторого опорного плана X . Соответствующие расширенные векторы $\bar{A}_1, \dots, \bar{A}_m$ также линейно независимы. Обозначим через Π гиперплоскость, натянутую на векторы $\bar{A}_1, \dots, \bar{A}_m$ и проходящую через начало координат. Эта гиперплоскость однозначно определяется опорным планом X и является его образом в пространстве точек U . Гиперплоскость Π делит расширенные векторы $\bar{A}_{m+1}, \dots, \bar{A}_n$ на две группы. Векторы условий первой группы расположены по ту же сторону от Π , что и положительная полуось Ou_{m+1} . Мы будем говорить, что эти векторы лежат «выше» гиперплоскости Π . Другая группа содержит все векторы \bar{A}_j , лежащие «ниже» Π .

Будем, как обычно, одним и тем же символом обозначать как вектор, так и определяемую им точку. Пусть прямые, проходящие через точки A_j и \bar{A}_j (параллельно Ou_{m+1}), пересекают гиперплоскость Π в точках A_j^0 , а прямая Q — в точке B^0 .

Выразим длины отрезков $A_j^0 A_j$, $\bar{A}_j A_j^0$ и $B^0 B$ через параметры метода последовательного улучшения плана. Это позволит нам связать геометрические построения с приведенным ранее алгебраическим описанием метода.

Точка A_j^0 по построению принадлежит гиперплоскости Π . Запишем разложение вектора A_j^0 по системе m линейно независимых расширенных векторов условий \bar{A}_i , порождающих гиперплоскость Π ,

$$A_j^0 = \sum_{i=1}^m \tilde{x}_{ij} \bar{A}_i. \quad (3.1)$$

Первые m компонент векторов A_j^0 и \bar{A}_i определяют векторы условий A_j и A_i соответственно. Для них имеем (см. (1.4))

$$A_j = \sum_{i=1}^m x_{ij} A_i.$$

Поэтому

$$\tilde{x}_{ij} = x_{ij}. \quad (3.2)$$

«Длина» направленного отрезка $A_j^0 A_j$ представляет собой $(m+1)$ -ю компоненту вектора A_j^0 (*). Последняя компонента расширенного вектора условий \bar{A}_i равна c_i . Поэтому из формул (3.1), (3.2) и (1.5) получаем

$$|A_j^0 A_j| = \sum_{i=1}^m x_{ij} c_i = z_j \quad (3.3)$$

и

$$|\bar{A}_j A_j^0| = |\bar{A}_j A_j| - |A_j^0 A_j| = c_j - z_j = -\Delta_j. \quad (3.4)$$

Отрезок $B^0 B$ определяет значение линейной формы для рассматриваемого опорного плана

$$|B^0 B| = L(X). \quad (3.5)$$

Таким образом, для расширенных векторов первой группы, проходящих «выше» гиперплоскости Π , величины $\Delta_j < 0$. Для векторов \bar{A}_j второй группы $\Delta_j > 0$.

Пусть \bar{A}_j — произвольный расширенный вектор условий, не принадлежащий гиперплоскости Π . Рассмотрим $(m+1)$ -мерный конус, образованный векторами $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_m, \bar{A}_j$. Обозначим его через K_j (индекс j — номер вектора $\bar{A}_j \notin \Pi$). Точки общей части прямой Q и конуса K_j являются геометрическими образами планов, которые образуются из X путем элементарного преобразования, связанного с вектором A_j ($0 \leq \theta \leq \theta_0$). В невырожденном случае точка X лежит внутри грани, образованной векторами $\bar{A}_1, \dots, \bar{A}_m$. Поэтому общая часть Q и K_j при любом $\bar{A}_j \notin \Pi$ содержит некоторый отрезок. В вырожденном случае общая часть Q и K_j может содержать лишь точку X (Q касается конуса K_j).

Первый этап итерации состоит в проверке опорного плана на оптимальность. Геометрически проверка сводится к решению вопроса, имеются ли расширенные векторы, расположенные «над» гиперплоскостью Π . Если первая группа расширенных векторов — пустое множество, опорный план X является оптимальным (случай 1°).

*) Здесь под «длиной» $|AB|$ направленного отрезка AB подразумевается длина отрезка AB с учетом знака. $|AB| > 0$ ($|AB| < 0$), если A «выше» («ниже») B в смысле оси Ou_{m+1} .

В этом случае весь конус K расположен под гиперплоскостью Π . Максимальное значение линейной формы определяется «длиной» отрезка B^0B .

Пусть теперь первая группа векторов \bar{A}_j не пуста, т. е. имеются расширенные векторы условий, расположенные «над» гиперплоскостью Π . Здесь представляются две возможности, которые приводят к случаям 2° и 3° соответственно.

Случай 2° геометрически означает существование такого вектора \bar{A}_j , расположенного «над» гиперплоскостью Π , что общая часть K_j и Q является лучом, исходящим из точки X . Заметим, что это возможно в том и только в том случае, когда конус K_j содержит полуось Ou_{m+1} . Если первая группа расширенных векторов условий не пуста, и любой конус K_j не содержит Ou_{m+1} , то имеем дело со случаем 3°. В случае 3° можно перейти от гиперплоскости Π к гиперплоскости Π' , соответствующей очередному опорному плану X .

Гиперплоскость Π' пересекает прямую Q в точке B' , расположенной «выше» B^0 . Переход от Π к Π' осуществляется следующим образом. Пусть вектор \bar{A}_k лежит «выше» гиперплоскости Π . Образует конус K_k . Как отмечалось, в невырожденном случае общая часть K_k и Q содержит некоторый отрезок. В случае 3° общая часть K_k и Q не может оказаться лучом. Поэтому она совпадает с отрезком, концами которого являются «нижняя» и «верхняя» точки пересечения K_k и Q . Точка B^0 — «нижняя» точка отрезка. «Верхнюю» точку отрезка обозначим через B' . Заметим, что $(m+1)$ -мерный конус K_k имеет $m+1$ граней размерности m . Каждая из них является конусом, образованным некоторыми m векторами системы $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_m, \bar{A}_k$. Обозначим грань конуса K_k , не содержащую вектора \bar{A}_j ($j=1, 2, \dots, m, k$), через G_{jk} . В невырожденном случае точка B' расположена внутри некоторой грани конуса K_k , скажем G_{rk} .

Пусть Π' — гиперплоскость, содержащая грань G_{rk} . Гиперплоскость Π' является геометрическим образом нового опорного плана X' . Очевидно, базис плана X' состоит из векторов $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_{r-1}, \bar{A}_{r+1}, \dots, \bar{A}_m$.

\bar{A}_k , участвующих в образовании грани Γ_{rk} . Прямая Q пересекает гиперплоскость Π' в точке B' . Точка B' лежит «выше» B^0 , следовательно, план X' соответствует большему значению линейной формы задачи по сравнению с планом X . На этом заканчивается итерация метода. Следующая итерация ведется по тем же правилам. Снова проверяем, лежит ли гиперплоскость Π' «выше» конуса K или нет. В первом случае гиперплоскость Π' определяет оптимальный план, а точка B' — максимальное значение линейной формы. Во втором случае следует перейти от гиперплоскости Π' к гиперплоскости Π'' по тем же правилам, по которым был совершен переход от Π к Π' . В невырожденной задаче мы, таким образом, всегда придем к гиперплоскости Π^* , расположенной «над» конусом K , либо убедимся в неразрешимости задачи.

§ 4. ВЫРОЖДЕННОСТЬ

4.1. В ряде предыдущих рассуждений было использовано допущение о невырожденности задачи линейного программирования.

Используя геометрическую интерпретацию задачи линейного программирования для простейшего плоского случая, легко отличить вырожденную задачу линейного программирования от невырожденной. В вырожденной задаче в одной вершине многоугольника пересекается более двух прямых, описываемых уравнениями вида $x_k = 0$. Это значит, что одна или несколько сторон многоугольника условий стягиваются в точку. Аналогия может быть продолжена и дальше.

В терминах второй геометрической интерпретации задачи линейного программирования вырожденность (в случае $m = 2$) означает, что прямая Q пересекает некоторое ребро многогранного конуса. При $m > 2$ вырожденность означает, что некоторая общая точка прямой Q и конуса задачи принадлежит многогранному конусу, порожденному не более чем $(m - 1)$ ребром. Векторы условий при положительных компонентах невырожденного опорного плана однозначно определяют базис опорного плана. В вырожденной задаче некоторые опорные планы содержат $v < m$ положительных составляющих.

В соответствии с определением базис такого плана образуется из ν векторов, отвечающих положительным компонентам, и $(m - \nu)$ векторов, связанных с нулевыми составляющими плана. Очевидно, что в этом случае базис определяется неоднозначно. Единственное требование, предъявляемое к дополнительным $(m - \nu)$ векторам, состоит в том, чтобы полученная система из m векторов была линейно независима.

Предположение о невырожденности задачи использовалось в предыдущих параграфах дважды: при вычислении значения θ_0 и при выборе вектора, подлежащего исключению из базиса. Рассмотрим каждый из этих случаев.

При вырожденности плана X величина θ_0 может оказаться равной нулю. Это будет при $x_{i0} = x_{si} = 0$ и $x_{ik} > 0$. Если $\theta_0 = 0$, то новый опорный план X' совпадает со старым планом X . В результате итерации меняется лишь базис этого плана.

В предположении о невырожденности существует только одно значение $i_r^*(i=r)$, при котором достигается $\min_i (x_{i0}/x_{ik})$ для $x_{ik} > 0$. Это позволяет однозначно определить вектор A_r , который следует исключить из базиса и заменить вектором A_k . Приведенные рассуждения теряют силу, если окажется, что существует несколько значений i , при которых x_{i0}/x_{ik} (для $x_{ik} > 0$) обращается в минимум. В этом случае при $\theta_0 = \min_{x_{ik} > 0} (x_{i0}/x_{ik})$ в раз-

ложении (2.4) обратится в нуль несколько коэффициентов при векторах условий. Один из этих векторов должен быть исключен из базиса.

Если в течение ряда последовательных итераций величина θ_0 равна нулю, то процесс решения задачи для этих итераций состоит в движении по базисам одного и того же опорного плана. Поскольку значение линейной формы задачи, естественно, остается неизменным, нет основания утверждать, что движение осуществляется по различным базисам. В невырожденной задаче только изменение линейной формы гарантировало от повторения базиса. В вырожденной задаче появляется возможность цикла — возможность возвращения к базису, который уже встречался.

Зацикливание является редким событием в практике линейного программирования. Построение примера вырожденной задачи, приводящей к зацикливанию, вызвало известные затруднения.

Первый опубликованный пример цикла принадлежит Билу [2]. В литературе встречаются также ссылки на неопубликованный пример цикла, построенный ранее Гофманом [13].

Приведем искусственным образом построенный пример задачи линейного программирования, приводящей к зацикливанию. Требуется обратить в максимум линейную форму

$$L(X) = x_3 - x_4 + x_5 - x_6$$

при условиях

$$x_1 + x_3 - 2x_4 - 3x_5 + 4x_6 = 0,$$

$$x_2 + 4x_3 - 3x_4 - 2x_5 + x_6 = 0,$$

$$x_3 + x_4 + x_5 + x_6 + x_7 = 1,$$

$$x_j \geq 0, \quad j = 1, 2, \dots, 7.$$

Цикл получается при следующей последовательности решения задачи, не противоречащей рекомендациям предыдущих параграфов. В базис вводятся последовательно векторы A_3, A_4, A_5, A_6, A_1 и A_2 , которые помещаются соответственно в 1-ю, 2-ю, 1-ю, 2-ю, 1-ю, 2-ю позиции базиса. При выборе позиции, в которую следует поместить включаемый в базис вектор, неоднозначность имела место на 1-м, 3-м и 5-м шагах. Линейная форма при всех шести итерациях не меняет своей величины. После шести шагов возвращаемся к исходному базису.

Приведем без доказательства некоторые утверждения, полученные при построении примера. Жесткость перечисленных ниже, необходимых, но явно недостаточных условий зацикливания объясняет, в частности, почему цикл в задачах линейного программирования — столь редко встречающееся явление.

Зацикливание невозможно, если вектор ограничений выражается не менее чем через $(m-1)$ векторов условий. Цикл может встретиться только в задачах, в которых возможно разложение вектора ограничений по $(m-2)$

или менее векторам условий. При этом цикл не может включать менее шести шагов. Тем не менее принципиальная возможность цикла требует практических рекомендаций, обеспечивающих во всех случаях решение вырожденных задач линейного программирования.

4.2. Геометрическое истолкование явления вырожденности указывает пути устранения опасности заикливания. Вырожденность, как уже отмечалось, соответствует тому случаю, когда прямая Q пересекает хотя бы один из многогранных конусов, порожденных не более чем $(m-1)$ ребром. Следует ожидать, что, смещая несколько прямую Q параллельно самой себе, можно исключить этот случай. Смещение прямой Q связано с изменением вектора ограничений, однозначно определяющего ее положение. Итак, небольшое соответствующим образом подобранное изменение вектора ограничений может устранить опасность возникновения цикла. На этом основана идея приводимой ниже рекомендации, гарантирующей во всех случаях решение задачи линейного программирования за конечное число шагов.

Пусть базис составлен из векторов условий A_1, A_2, \dots, A_m . Разложение вектора ограничений и векторов условий по векторам базиса может быть представлено в виде

$$B = A_1 x_1 + \dots + A_m x_m, \quad (4.1)$$

$$A_j = A_1 x_{1j} + \dots + A_m x_{mj}. \quad (4.2)$$

Осуществим сдвиг вектора ограничений следующим образом. Умножим (4.2) на ε^j ($\varepsilon > 0$), просуммируем полученные соотношения по j от 1 до n и сложим результат с соотношением (4.1). Получим

$$B + \sum_{j=1}^n \varepsilon^j A_j = A_1 \left(x_1 + \sum_{j=1}^n \varepsilon^j x_{1j} \right) + \dots \\ \dots + A_m \left(x_m + \sum_{j=1}^n \varepsilon^j x_{mj} \right).$$

Введем обозначения:

$$R(\varepsilon) = B + \sum_{j=1}^n \varepsilon^j A_j, \quad x_i(\varepsilon) = x_i + \sum_{j=1}^n \varepsilon^j x_{ij}.$$

С каждой задачей линейного программирования (1.1)—(1.3) свяжем семейство так называемых ε -задач. В ε -задаче вычисляется экстремум линейной формы (1.1) при условиях (1.2), в которых вектор B заменен на $B(\varepsilon)$, и (1.3).

Правило исключения вектора из базиса, гарантирующее от заикливания в процессе решения, может быть выработано на основе следующих двух утверждений:

I. Существует такое малое $\varepsilon_1 > 0$, что для любого $0 < \varepsilon < \varepsilon_1$ ε -задача является невырожденной.

II. Существует такое малое $\varepsilon_2 > 0$, что для любого $0 \leq \varepsilon < \varepsilon_2$ всякому опорному плану ε -задачи соответствует опорный план исходной задачи. При этом оптимальному плану ε -задачи соответствует решение исходной задачи. (Опорные планы исходной задачи и ε -задачи называются соответствующими, если они имеют общий базис.)

Доказательства этих утверждений, необходимых для теоретических применений метода последовательного улучшения плана, но несущественных для вычислительных схем, мы здесь опускаем. Подробное и достаточно строгое исследование различных аспектов явления вырожденности приведено в [50].

4.3. Утверждения I и II предыдущего пункта служат теоретической основой для разработки правила, устраняющего неоднозначность при исключении вектора из базиса вырожденного опорного плана и дающего полную гарантию от возможности образования цикла.

Вместо исходной задачи рассматривается ε -задача при $0 < \varepsilon < \varepsilon_0$. В качестве $\varepsilon_0 > 0$ выбирается число, не превышающее меньшего из двух чисел ε_1 и ε_2 . В силу утверждения I ($\varepsilon < \varepsilon_1$) рассматриваемая ε -задача не вырождена. Поэтому метод последовательного улучшения плана приведет через конечное число шагов к решению задачи. В силу утверждения II ($\varepsilon < \varepsilon_2$) базис полученного решения оказывается вместе с тем базисом оптимального плана исходной задачи. Если решение ε -задачи записано в форме

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

то для получения решения исходной задачи достаточно положить ε равным нулю. Если же оптимальный план ε -задачи получен для фиксированного значения ε , то для получения решения исходной задачи следует разложить вектор ограничений по базису оптимального плана ε -задачи.

Заметим, что практически в определении ε_0 нет необходимости. Существование ε_0 важно лишь для обоснования правила решения вырожденных задач. При использовании ε -приема можно считать ε меньшим любого из положительных чисел, с которым его приходится сравнивать в процессе решения задачи.

На основе приведенных соображений можно указать правило удаления вектора из базиса, исключающее возможность заикливания. В вырожденной задаче условие

$$\theta = \min_i \frac{x_{i0}}{x_{ik}}, \quad x_{ik} > 0, \quad (4.3)$$

может осуществляться сразу для нескольких значений i . Отсутствие правила, указывающего, какой из векторов A_i нужно при этом выводить из базиса и какой следует оставить,—основное затруднение процедуры. Приведенные выше рассуждения снимают эти трудности. Производя ε -сдвиг вектора ограничений, перепишем условие (4.3) в виде

$$\begin{aligned} \theta &= \min_{\substack{i \\ x_{ik} > 0}} \frac{x_{i0}(\varepsilon)}{x_{ik}} = \\ &= \min_{\substack{i \\ x_{ik} > 0}} \left[\frac{x_{i0}}{x_{ik}} + \varepsilon \frac{x_{i1}}{x_{ik}} + \varepsilon^2 \frac{x_{i2}}{x_{ik}} + \dots + \varepsilon^n \frac{x_{in}}{x_{ik}} \right]. \end{aligned} \quad (4.4)$$

Здесь x_{i0}/x_{ik} —отношения составляющих вектора $A_0 = B$ к соответствующим компонентам вектора A_k , вводимого в базис; x_{i1}/x_{ik} —отношения составляющих векторов A_1 и A_k и т. д.

Пусть $\min_{x_{ik} > 0} (x_{i0}/x_{ik})$ достигается для нескольких значений индекса i . Чтобы выбрать из них индекс вектора, который должен быть изъят из базиса, следует сравнить для них значения x_{i1}/x_{ik} . Вектор A_r , для которого x_{i1}/x_{ik}

принимает наименьшее значение, и является искомым. Может оказаться, что не только x_{i0}/x_{ik} , но и x_{i1}/x_{ik} достигает минимума при нескольких (общих для обоих отношений) значениях i . Тогда процесс сравнения отношений компонент очередного вектора условий и соответствующих компонент A_k должен быть продолжен. Таким образом, всегда удастся однозначно выделить вектор A_r , который должен быть удален из базиса и заменен вектором A_k . Приведенное правило выделения вектора A_r , подлежащего выводу из базиса, вытекает из формулы (4.4). Если исходная задача является вырожденной, то переход от одного базиса к следующему может не сопровождаться увеличением линейной формы. Однако в силу доказанной невырожденности ε -задачи (при достаточно малых ε) заикливание уже невозможно. В процессе решения задачи исключается возвращение к уже пройденному базису и решение получается за конечное число шагов.

Как мы видели, вырожденный случай не вносит никаких особенностей в правила решения задач методом последовательного улучшения плана. Необходимо лишь сохранить порядок столбцов в таблицах алгоритма. Любая нумерация векторов A_j может быть принята за исходную, но этот порядок должен сохраниться во всех дальнейших вычислениях. (Строго говоря, порядок векторов необходимо сохранять лишь между двумя последовательными изменениями линейной формы.)

Применим приведенное правило к примеру задачи с циклом из п. 4.1. На первом шаге, когда в базис вводится вектор A_3 , величина

$$\theta_0 = \frac{x_{10}}{x_{13}} = \frac{x_{20}}{x_{23}} = 0$$

достигается на первых двух позициях базиса.

Ранее вектор A_3 был помещен в 1-ю позицию базиса. В соответствии с правилом, гарантирующим от цикла, следует сравнить отношения x_{11}/x_{13} и x_{21}/x_{23} .

Поскольку

$$\frac{x_{11}}{x_{13}} = 1 > \frac{x_{21}}{x_{23}} = 0,$$

вектор A_3 следует поместить во вторую позицию базиса. Нетрудно проверить, что, поступив таким образом, мы уже на следующей итерации увеличим значение линейной формы.

Как отмечалось, заикливание в задачах линейного программирования — редкое явление. До сих пор ни одна практическая задача линейного программирования не привела к циклу, хотя при решении подавляющего большинства из них не использовалось строгое правило выбора вектора, исключаемого из базиса. Поэтому естественно в практических расчетах заменить приведенное здесь правило более простым. В частности, можно рекомендовать следующее упрощенное правило.

Из базиса исключается тот из «подозрительных» векторов условий, который имеет наименьший номер. Если при этом будет получен цикл, то, начиная от базиса, полученного вторично, следует применять строгое правило выбора вектора A_r , подлежащего исключению из базиса. После того как линейная форма увеличится, целесообразно снова вернуться к упрощенному правилу.

§ 5. СВЯЗЬ МЕЖДУ ПАРАМЕТРАМИ ПОСЛЕДОВАТЕЛЬНЫХ ИТЕРАЦИЙ

5.1. Связи между параметрами соседних итераций для разных алгоритмов конечных методов линейного программирования устанавливаются по одному и тому же принципу. Ниже доказывается утверждение, из которого будут далее получены в качестве следствий все рекуррентные формулы, необходимые для решения задачи по различным вычислительным схемам конечных методов линейного программирования.

Рассмотрим две системы линейно независимых векторов $\{P_i\}_{i \in I}$ и $\{P_i\}_{i \in I'}$. Множества индексов I и I' различаются только одним элементом. Множество I' образуется из I заменой индекса r на индекс k , так что

$$\{I_{i \neq r}\} \equiv \{I'_{i \neq k}\}. \quad (5.1)$$

Обозначим коэффициенты разложения некоторой системы векторов Q_j по векторам двух систем линейно

независимых векторов $\{P_i\}_{i \in I}$ и $\{P_i\}_{i \in I'}$ соответственно через ξ_{ij} и ξ'_{ij} , т. е.

$$Q_j = \sum_{i \in I} \xi_{ij} P_i, \quad (5.2)$$

$$Q_j = \sum_{i \in I'} \xi'_{ij} P_i. \quad (5.3)$$

Будем считать известными коэффициенты η_{ik} разложения вектора P_k по векторам системы $\{P_i\}_{i \in I}$:

$$P_k = \sum_{i \in I} \eta_{ik} P_i. \quad (5.4)$$

Здесь $\eta_{rk} \neq 0$, так как обе системы $\{P_i\}_{i \in I}$ и $\{P_i\}_{i \in I'}$ состоят из линейно независимых векторов и различаются только одним вектором P_r (P_k).

Установим связь между ξ'_{ij} , с одной стороны, и ξ_{ij} и η_{ik} , с другой. Имеет место следующее утверждение.

Теорема 5.1. Коэффициенты ξ_{ij} и ξ'_{ij} разложения произвольных векторов Q_j по векторам систем $\{P_i\}_{i \in I}$ и $\{P_i\}_{i \in I'}$, для которых имеют место соотношения (5.1) и (5.4), связаны между собой формулами

$$\xi'_{ij} = \begin{cases} \xi_{ij} - \frac{\xi_{rj}}{\eta_{rk}} \eta_{ik} & \text{при } i \neq k, \\ \frac{\xi_{rj}}{\eta_{rk}} & \text{при } i = k. \end{cases} \quad (5.5)$$

Доказательство. Из равенства (5.4) имеем

$$P_k = \sum_{\substack{i \in I \\ i \neq r}} \eta_{ik} P_i + \eta_{rk} P_r.$$

По условию $\eta_{rk} \neq 0$. Поэтому, используя формулы (5.1) и (5.2), приходим к соотношению

$$\begin{aligned} Q_j &= \sum_{\substack{i \in I \\ i \neq r}} \xi_{ij} P_i + \frac{\xi_{rj}}{\eta_{rk}} \left(P_k - \sum_{\substack{i \in I \\ i \neq r}} \eta_{ik} P_i \right) = \\ &= \sum_{\substack{i \in I' \\ i \neq k}} \left(\xi_{ij} - \frac{\xi_{rj}}{\eta_{rk}} \right) P_i + \frac{\xi_{rj}}{\eta_{rk}} P_k. \end{aligned}$$

Система векторов $\{P_i\}_{i \in I'}$ линейно независима. Поэтому коэффициенты разложения вектора Q_j по векторам этой

системы определяются однозначно. Сравнивая последнее соотношение с формулой (5.3), убеждаемся в справедливости утверждения теоремы.

Вектор P_r в системе $\{P_i\}_{i \in I'}$ занимает то же место, что и вектор P_r в системе $\{P_i\}_{i \in I}$. Мы уже видели, что в ряде случаев индексы компонент удобно связывать не с номерами соответствующих векторов, а с позициями, занимаемыми векторами в системе. В этих предположениях целесообразно переписать рекуррентные формулы (5.5) в виде

$$\xi'_{ij} = \begin{cases} \xi_{ij} - \frac{\xi_{rj}}{\eta_{rk}} \eta_{ik} & \text{при } i \neq r, \\ \frac{\xi_{rj}}{\eta_{rk}} & \text{при } i = r. \end{cases} \quad (5.6)$$

Здесь мы предполагаем, что вектор P_r занимает r -ю позицию в системе $\{P_i\}_{i \in I}$.

5.2. Воспользуемся формулами (5.6) для вывода рекуррентных соотношений, связывающих параметры соседних итераций в первом алгоритме метода последовательного улучшения плана.

В первом алгоритме, связанном с первой формой признака оптимальности опорного плана, необходимо на каждом шаге метода вычислять следующие параметры:

а) x_{ij} — составляющие разложения произвольного вектора условий A_j ($j=1, 2, \dots, n$) по векторам базиса;

б) x_{i0} — базисные компоненты опорного плана ($i=1, 2, \dots, m$);

в) Δ_j — оценки векторов условий A_j относительно рассматриваемого базиса ($j=1, 2, \dots, n$);

г) $L(X)$ — значение линейной формы на плане X .

Введем обозначение

$$V = -\bar{e}_{m+1},$$

где $\bar{e}_{m+1} = (0, 0, \dots, 0, 1)$ есть $(m+1)$ -мерный единичный вектор.

Примем

$$\{P_i\}_{i \in I} = \{(\bar{A}_i)_{i \in I_X}, V\}, \quad (5.7)$$

$$\{P_i\}_{i \in I'} = \{(\bar{A}_i)_{i \in I_X}, V\}. \quad (5.8)$$

Приведенная запись означает, что в первую систему включены m расширенных векторов условий, определяемых базисом опорного плана X , и $(m+1)$ -мерный вектор V . Вторая система векторов состоит из расширенных векторов, отвечающих опорному плану X' , и вектора V . Опорный план X' получен из опорного плана X элементарным преобразованием, связанным с вектором A_k . Обе системы различаются только одним вектором. Вектор \bar{A}_k во второй системе занимает r -ю позицию базиса. В первой системе на этом месте расположен вектор A_{s_r} . Ясно, что как векторы первой системы, так и векторы второй системы линейно независимы.

Примем в качестве вектора Q_j произвольный расширенный вектор условий \bar{A}_j , $j=0, 1, 2, \dots, n$. (Под расширенным вектором \bar{A}_0 здесь подразумевается $(m+1)$ -мерный вектор, первые m компонент которого совпадают с соответствующими составляющими вектора ограничений, а $(m+1)$ -я компонента c_0 равна нулю.)

Разложение вектора \bar{A}_j по векторам первой системы может быть записано в следующем виде:

$$\bar{A}_j = \sum_{i=1}^m \zeta_{ij} \bar{A}_{s_i} + \zeta_{m+1,j} V, \quad j=0, 1, 2, \dots, n. \quad (5.9)$$

Аналогично записывается разложение вектора \bar{A}_k (играющего роль вектора P_k в теореме 5.1):

$$\bar{A}_k = \sum_{i=1}^m \eta_{ik} \bar{A}_{s_i} + \eta_{m+1,k} V. \quad (5.10)$$

Первые m компонент расширенных векторов определяют векторы условий и связаны соотношениями

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}, \quad j=0, 1, 2, \dots, n,$$

и соответственно

$$A_k = \sum_{i=1}^m x_{ik} A_{s_i}.$$

Учитывая, что первые m компонент вектора V равны нулю, получаем

$$\begin{aligned}\zeta_{ij} &= x_{ij}, & i &= 1, 2, \dots, m, & j &= 0, 1, \dots, n, \\ \eta_{ik} &= x_{ik}, & i &= 1, 2, \dots, m.\end{aligned}$$

Аналогичные рассуждения приводят к равенствам

$$\zeta'_{ij} = x'_{ij}, \quad i = 1, 2, \dots, m, \quad j = 0, 1, \dots, n.$$

Чтобы вычислить $\zeta_{m+1, j}$ и $\eta_{m+1, k}$ для $j = 0, 1, \dots, n$, спроектируем обе части равенств (5.9) и (5.10) на ось \bar{e}_{m+1} или, что то же самое, вычислим $(m+1)$ -е составляющие векторов из левой и правой частей каждого из этих равенств:

$$c_j = \sum_{i=1}^m x_{ij} c_{s_i} - \zeta_{m+1, j}, \quad c_k = \sum_{i=1}^m x_{ik} c_{s_i} - \eta_{m+1, k}.$$

Сравнивая полученные формулы с формулой (2.1) и учитывая, что $c_0 = 0$, приходим к соотношениям

$$\begin{aligned}\zeta_{m+1, j} &= \Delta_j, & j &= 1, 2, \dots, n, & \eta_{m+1, k} &= \Delta_k, \\ \zeta_{m+1, 0} &= \Delta_0 = L(X).\end{aligned}$$

Точно так же

$$\begin{aligned}\zeta'_{m+1, j} &= \Delta'_j, & j &= 1, 2, \dots, n, \\ \zeta'_{m+1, 0} &= \Delta'_0 = L(X').\end{aligned}$$

Естественно ввести следующие обозначения:

$V = \bar{A}_{s_{m+1}}$, $\Delta_j = x_{m+1, j}$, $j = 1, 2, \dots, n$, $\Delta_0 = L(X) = x_{m+1, 0}$ и соответственно

$$\Delta'_j = x'_{m+1, j}, \quad j = 1, 2, \dots, n, \quad \Delta'_0 = L(X') = x'_{m+1, 0}.$$

В новых обозначениях формулы (5.9) и (5.10) принимают такой вид:

$$\left. \begin{aligned}\bar{A}_j &= \sum_{i=1}^{m+1} x_{ij} \bar{A}_{s_i}, & j &= 0, 1, \dots, n, \\ \bar{A}_k &= \sum_{i=1}^{m+1} x_{ik} \bar{A}_{s_i}.\end{aligned} \right\} \quad (5.11)$$

Из определения элементарного преобразования следует, что $\eta_{rk} = x_{rk} \neq 0$. (Индекс r — это номер позиции базиса, на которой достигается минимум отношения x_{i0}/x_{ik} при $x_{ik} > 0$.) Таким образом, выполняются все условия теоремы 5.1. Применяя утверждения этой теоремы к составляющим разложения расширенных векторов условий по векторам систем (5.7) и (5.8), получаем следующий результат.

Параметры двух последовательных итераций связаны рекуррентными формулами

$$x'_{ij} = \begin{cases} x_{ij} - \frac{x_{rf}}{x_{rk}} x_{ik} & \text{при } i \neq r, \\ \frac{x_{rf}}{x_{rk}} & \text{при } i = r, \end{cases} \quad (5.12)$$

$$i = 1, 2, \dots, m+1, \quad j = 0, 1, 2, \dots, n.$$

Соотношения (5.12) полностью обеспечивают переход от одного шага метода последовательного улучшения плана к следующему. Рекуррентные формулы (5.12) позволяют вычислять следующие параметры каждой итерации (если известны параметры предыдущего приближения):

- а) составляющие разложения x_{ij} всех векторов условий по векторам базиса ($i = 1, 2, \dots, m, j = 1, 2, \dots, n$);
- б) базисные компоненты опорного плана x_{i0} ($i = 1, 2, \dots, m, j = 0$);

в) относительные оценки векторов условий $\Delta_j = x_{m+1,j}$ ($i = m+1, j = 1, 2, \dots, n$);

г) значение линейной формы $L(X) = x_{m+1,0}$ ($i = m+1, j = 0$).

Формулы (5.12) являются основой первого алгоритма метода последовательного улучшения плана.

5.3. Используем теперь формулы (5.6) для вывода рекуррентных соотношений, связывающих параметры последовательных итераций во втором алгоритме метода последовательного улучшения плана.

Второй алгоритм основан на втором признаке оптимальности опорного плана, в котором оценки векторов условий Δ_j вычисляются на каждом шаге с помощью

вектора $\Lambda = (\lambda_1, \dots, \lambda_m)$. Параметры λ_i в свою очередь удовлетворяют системе уравнений

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j, \quad j \in I_X. \quad (5.13)$$

Как видим, вычисление λ_i связано с обращением матрицы A_X . Для решения задач вторым алгоритмом необходимо вычислять на каждом шаге следующие параметры:

- а) e_{ij} — элементы обратной матрицы A_X^{-1} ;
- б) λ_i — оценки условий задачи относительно базиса опорного плана X ;
- в) x_{i0} — базисные компоненты опорного плана;
- г) $L(X)$ — значение линейной формы на плане X .

Заметим, что параметры e_{ij} — элементы матрицы A_X^{-1} — удобно рассматривать как коэффициенты разложения m -мерных единичных векторов

$$e_j = \left(\underbrace{0, \dots, 0}_{j}, \overbrace{1, 0, \dots, 0}^m \right)$$

по векторам базиса.

Примем, как и в предыдущем пункте, в качестве систем векторов $\{P_i\}_{i \in I}$ и $\{P_i\}_{i \in I'}$ системы (5.7) и (5.8). Под векторами Q_j будем здесь подразумевать $(m+1)$ -мерные единичные векторы

$$\bar{e}_j = \left(\underbrace{0, \dots, 0}_j, \overbrace{1, 0, \dots, 0}^{m+1} \right), \quad j = 1, 2, \dots, m \text{ и}$$

$$\bar{e}_0 = \bar{A}_0 = (b_1, \dots, b_m, 0).$$

Запишем разложение вектора \bar{e}_j по векторам первой системы:

$$\bar{e}_j = \sum_{i=1}^m \xi_{ij} \bar{A}_{s_i} + \xi_{m+1,j} V, \quad j = 0, 1, 2, \dots, m. \quad (5.14)$$

Разложение m -мерных векторов e_j по векторам базиса (A_{s_i}) по определению представляется в виде

$$e_j = \sum_{i=1}^m e_{ij} A_{s_i}, \quad j = 0, 1, 2, \dots, m.$$

Следовательно,

$$\zeta_{ij} = e_{ij}, \quad i = 1, 2, \dots, m, \quad j = 0, 1, \dots, m$$

($e_{i0} = x_{i0}$ — базисные компоненты опорного плана).

Точно так же

$$\zeta'_{ij} = e'_{ij}, \quad i = 1, 2, \dots, m, \quad j = 0, 1, \dots, m.$$

Чтобы вычислить $\zeta_{m+1,j}$ ($j = 1, 2, \dots, n$), спроектируем обе части равенства (5.17) на вектор \bar{e}_{m+1} . Получим

$$0 = \sum_{i=1}^m e_{ij} c_{s_i} - \zeta_{m+1,j}.$$

Умножим обе части последнего равенства на a_{js_t} и просуммируем по j от 1 до m .

Учитывая, что a_{ij} и e_{ij} ($i = 1, \dots, m, j = 1, \dots, m$) — элементы взаимно обратных матриц, приходим к соотношению

$$0 = c_{s_t} - \sum_{j=1}^m a_{js_t} \zeta_{m+1,j}, \quad t = 1, 2, \dots, m.$$

Заменим индекс суммирования j на i . Принимая во внимание, что индексы s_t ($t = 1, 2, \dots, m$) составляют множество I_X , получим

$$\sum_{i=1}^m a_{ij} \zeta_{m+1,i} = c_j, \quad j \in I_X.$$

Таким образом, вектор $\{\zeta_{m+1,i}\}_i$ является решением системы (5.13). Следовательно,

$$\zeta_{m+1,j} = \lambda_j, \quad j = 1, 2, \dots, m. \quad (5.15)$$

Положим в формуле (5.14) $j = 0$ и перепишем это равенство для $(m+1)$ -х компонент векторов \bar{A}_{s_i} и V . Получим

$$0 = \sum_{i=1}^m \zeta_{i0} c_{s_i} - \zeta_{m+1,0}.$$

Но $\zeta_{i0} = e_{i0} = x_{i0}$. Поэтому

$$\zeta_{m+1,0} = \sum_{i=1}^m x_{i0} c_{s_i} = L(X) = \lambda_0.$$

Естественно ввести обозначения

$$V = \bar{A}_{s_{m+1}}; \quad \lambda_j = e_{m+1,j}, \quad j = 1, \dots, m; \quad \lambda_0 = L(X) = e_{m+1,0}, \quad (5.16)$$

и соответственно

$$\lambda'_j = e'_{m+1, j}, \quad j = 1, \dots, m; \quad \lambda'_0 = L(X') = e'_{m+1, 0}.$$

Кроме того, мы приняли, что

$$x_{i0} = e_{i0}, \quad i = 1, 2, \dots, m, \quad (5.17)$$

и соответственно

$$x'_{i0} = e'_{i0}.$$

В новых обозначениях формула (5.14) принимает вид

$$\bar{e}_j = \sum_{i=1}^{m+1} e_{ij} \bar{A}_{s_i}, \quad j = 0, 1, 2, \dots, m. \quad (5.18)$$

Разложение векторов $\bar{e}_j = Q_j$ по векторам системы (5.8) может быть представлено аналогичной формулой

$$e_j = \sum_{i \in I'} e'_{ij} \bar{A}_{s'_i}, \quad j = 0, 1, 2, \dots, m. \quad (5.19)$$

Пусть $I = (s_1, s_2, \dots, s_r, \dots, s_m)$, $I' = (s'_1, s'_2, \dots, s'_r, \dots, s'_m)$,

$$s'_i = \begin{cases} s_i & \text{при } i \neq r, \\ k & \text{при } i = r. \end{cases}$$

Разложение вектора $\bar{A}_k = P_k$ по векторам первой системы записывается в соответствии с формулами (5.11) в виде

$$\bar{A}_k = \sum_{i=1}^{m+1} x_{ik} \bar{A}_{s_i}. \quad (5.20)$$

Применяя теперь теорему 5.1 к составляющим разложения векторов e_j по векторам систем (5.7) и (5.8) и учитывая формулы (5.18) – (5.20), получаем следующие рекуррентные формулы, связывающие параметры двух соседних итераций:

$$e'_{ij} = \begin{cases} e_{ij} - \frac{e_{rj}}{x_{rk}} x_{ik} & \text{при } i \neq r, \\ \frac{e_{rj}}{x_{rk}} & \text{при } i = r, \end{cases} \quad (5.21)$$

$$i = 1, 2, \dots, m+1, \quad j = 0, 1, 2, \dots, m.$$

Формулы (5.21) позволяют вычислять следующие значения параметров каждой итерации, если известны параметры предыдущего приближения:

а) элементы e_{ij} обратной матрицы A_X^{-1} векторов базиса ($i = 1, 2, \dots, m, j = 1, 2, \dots, m$);

б) базисные компоненты опорного плана $x_{i0} = e_{i0}$ ($i = 1, \dots, m, j = 0$);

в) относительные оценки условий задачи

$$\lambda_j = e_{m+1, j} \quad (i = m + 1, j = 1, 2, \dots, m);$$

г) значение линейной формы $L(X) = e_{m+1, 0}$ ($i = m + 1, j = 0$).

Рекуррентные формулы (5.21) являются основой второго алгоритма метода последовательного улучшения плана.

§ 6. ПЕРВЫЙ АЛГОРИТМ

6.1. Теоретические основы первого алгоритма были изложены в § 2. Здесь будет описан порядок вычислений при использовании этого алгоритма для решения задач линейного программирования, заданных в канонической форме.

Как уже отмечалось, метод последовательного улучшения плана позволяет, отправляясь от некоторого исходного опорного плана и постепенно улучшая его, получить через конечное число итераций оптимальный план или убедиться в неразрешимости задачи. Каждой итерации соответствует переход от одной таблицы алгоритма к следующей.

Каждая итерация метода состоит из двух этапов. Первый этап заключается в проверке исследуемого опорного плана на оптимальность. Второй этап проводится, если рассматриваемый план не оказался оптимальным и при этом не выявлена неразрешимость задачи. Второй этап заключается в определении элементарного преобразования, приводящего к новому опорному плану с большим значением линейной формы. На втором этапе определяется вектор, который должен быть введен в базис, и вектор, который должен быть исключен из базиса. После этого вычисляются базисные компоненты нового опорного плана и все параметры, необходимые для продолжения процесса решения задачи.

В новый базис может быть включен любой вектор, оценка которого относительно предшествующего базиса

отрицательна. Линейная форма получает на каждом шаге максимальное приращение, если в базис вводится вектор A_j , для которого $\theta_0^{(j)} \Delta_j$ принимает наименьшее значение. Опыт показывает, что при этом, как правило, уменьшается и число итераций, необходимое для решения задачи. Однако выявление вектора, обращающего в минимум произведение $\theta_0^{(j)} \Delta_j$, связано с достаточно громоздкими вычислениями. Многочисленные расчеты показывают, что обычно к решению задачи быстрее приводит более простой путь, когда в базис вводится вектор A_k с наименьшей оценкой Δ_k . При этом θ_0 в каждой итерации определяется только один раз и количество вычислений на каждом шаге заметно уменьшается. В настоящем параграфе вычислительная схема описывается применительно к указанному простому способу выбора вектора, включаемого в базис.

Выводу из базиса подлежит вектор, расположенный в позиции базиса, на которой достигается $\theta_0 = \min_{x_{ik} > 0} (x_{i0}/x_{ik})$

(k — номер вектора, включаемого в базис). Если θ_0 достигается на нескольких векторах, то из базиса исключается любой из них; для определенности можно исключать, например, вектор, отвечающий меньшему номеру позиции в базисе. Это простое правило не исключает возможности цикла. Однако заикливание в задачах линейного программирования — крайне редкое явление, и переход к более громоздкому правилу, гарантирующему от заикливания, следует рекомендовать лишь в случае, если цикл будет обнаружен. После устранения цикла целесообразно снова вернуться к упрощенному правилу определения вектора, подлежащего исключению из базиса.

После того как определен номер вектора, который должен быть включен в базис, и позиция базиса, в которой его следует расположить (позиция вектора, исключаемого из базиса), можно приступить к вычислению параметров очередного опорного плана. Определение базисных переменных нового опорного плана, коэффициентов разложения векторов условий по векторам нового базиса, оценок векторов условий относительно нового базиса и значения линейной формы на новом плане производится по рекуррентным формулам (5.12). Проведенные

таким образом расчеты полностью подготавливают следующую итерацию.

Результаты расчетов, отвечающих l -й итерации, удобно сводить в таблицу вида 4.1.

Таблица 4.1

N	C_X	B_X	A_0	A_1	A_2	\dots	A_k	\dots	A_n	θ
1	c_{s_1}	A_{s_1}	$x_{10}^{(i)}$	$x_{11}^{(i)}$	$x_{12}^{(l)}$	\dots	$x_{1k}^{(l)}$	\dots	$x_{1n}^{(l)}$	
2	c_{s_2}	A_{s_2}	$x_{20}^{(i)}$	$x_{21}^{(l)}$	$x_{22}^{(l)}$	\dots	$x_{2k}^{(l)}$	\dots	$x_{2n}^{(l)}$	
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
r	c_{s_r}	A_{s_r}	$x_{r0}^{(l)}$	$x_{r1}^{(l)}$	$x_{r2}^{(l)}$	\dots	$x_{rk}^{(l)}$	\dots	$x_{rn}^{(l)}$	θ_0
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
m	c_{s_m}	A_{s_m}	$x_{m0}^{(l)}$	$x_{m1}^{(l)}$	$x_{m2}^{(l)}$	\dots	$x_{mk}^{(l)}$	\dots	$x_{mn}^{(l)}$	
$m+1$	—	—	$L^{(l)}$	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	\dots	$\Delta_k^{(l)}$	\dots	$\Delta_n^{(l)}$	—

На рис. 4.1 изображена блок-схема отдельной итерации решения задачи линейного программирования по первому алгоритму метода последовательного улучшения плана.

Применение правила, позволяющего избежать заикливания в вырожденных задачах линейного программирования, требует окаймления справа таблиц итераций, отвечающих вырожденным планам, столбцами θ' , θ'' , \dots

В вырожденном случае $\theta_0 = \min_{x_{ik} > 0} (x_{i0}/x_{ik})$ достигается одновременно на ряде позиций. Поместим в столбец θ' отношения x_{i1}/x_{ik} для значений i , на которых достигается θ_0 . Выберем θ'_0 — наименьшее из этих чисел. Вектор, на котором достигается θ'_0 , подлежит исключению из базиса. Если таких векторов несколько, то описанная процедура

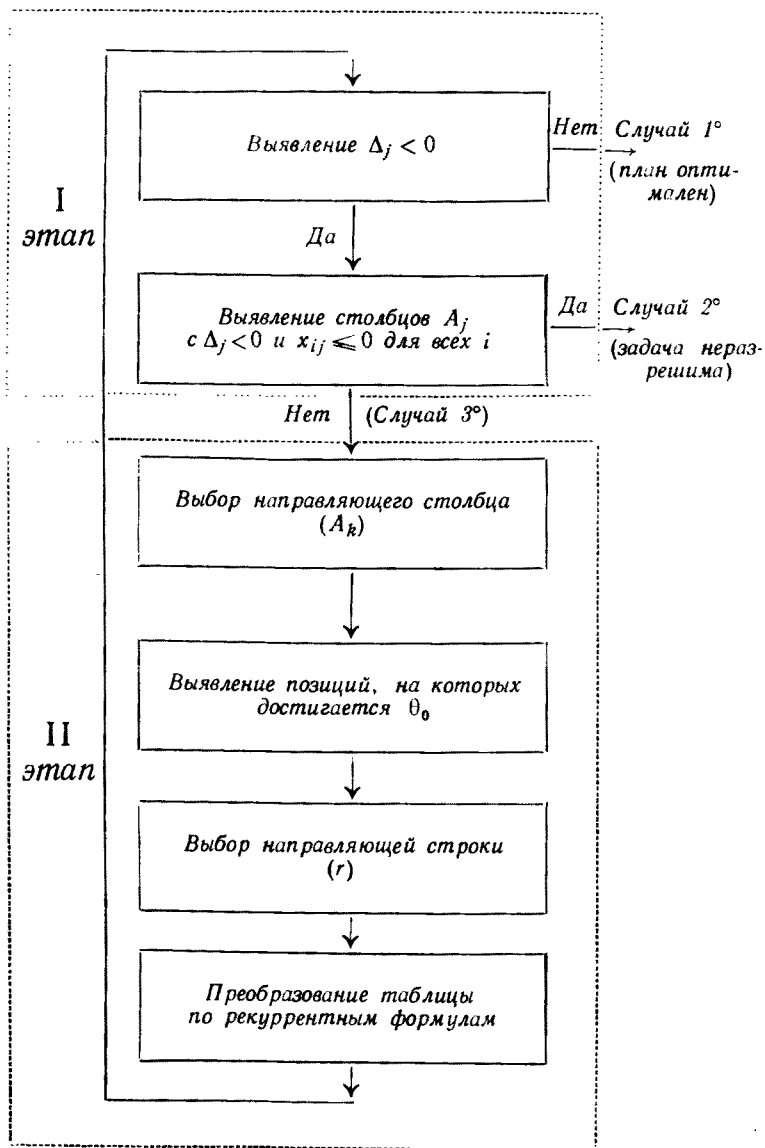


Рис. 4.1.

повторяется. В столбец θ помещаются отношения x_{i2}/x_{ik} для значений i , на которых достигается θ'_0 . Вычисления продолжаются до тех пор, пока позиция вектора, подлежащего исключению из базиса, не определится однозначно.

6.2. Приведем оценку трудоемкости вычислений по первому алгоритму метода последовательного улучшения плана.

Машинное время, необходимое для решения задачи, определяется главным образом числом умножений и делений, с которым связан процесс определения оптимального плана. Операции сравнения, сложения и вычитания проводятся на универсальных вычислительных машинах значительно быстрее, чем умножение и деление. Время обращения к памяти мы здесь не учитываем.

В первом алгоритме операция деления встречается дважды в каждой итерации: при заполнении строки таблицы, отвечающей вновь введенному в базис вектору ($x'_{rj} = x_{rj}/x_{rk}$, $j = 0, 1, 2, \dots, n$) и при вычислении столбца θ ($\theta = x_{i0}/x_{ik}$ при $x_{ik} > 0$). Заполнение строки x'_{rj} требует $n - m + 1$ делений, а для вычисления элементов столбца θ необходимо не более m делений. Таким образом, общее количество делений в каждой итерации первого алгоритма не превышает $n + 1$. Операции умножения встречаются в первом алгоритме только при преобразовании таблиц по рекуррентным формулам

$$x_{ij}^{(l+1)} = x_{ij}^{(l)} - \frac{x_{rj}^{(l)}}{x_{rk}^{(l)}} x_{ik}^{(l)} = x_{ij}^{(l)} - x_{rj}^{(l+1)} x_{ik}^{(l)}, \quad (6.1)$$

$$j = 1, 2, \dots, m, m + 1, \quad i \neq r, \quad j \neq 0, 1, 2, \dots, n.$$

При этом коэффициенты $x_{ij}^{(l+1)}$, соответствующие векторам базиса, не вычисляются (они равны нулю или единице). Каждая итерация требует, таким образом, $(n - m + 1)m$ умножений.

Количество итераций, необходимое для решения задачи линейного программирования, зависит от конкретных особенностей задачи и от начального опорного плана. Приемлемых теоретических оценок числа итераций для общей задачи линейного программирования пока не получено. Опыт показывает, что, как правило, число

итераций в различных задачах линейного программирования находится в пределах от m до $2m$.

Заметим, что для контрольных расчетов по первому алгоритму рекомендуется через определенное число итераций вычислять оценки Δ_j не только по рекуррентным формулам, но и непосредственно из соотношений

$$x_{m+1,j} = \sum_{i=1}^m x_{ij} c_{si} - c_j \quad (j=0, 1, 2, \dots, n). \quad (6.2)$$

Для этого требуется $m(n+1)$ умножение. Если не проверять оценки векторов базиса ($\Delta_j=0$ для $j \in I_X$), то дополнительное число умножений, с которым связан контроль, равно $m(n-m+1)$, т. е. столько же, сколько необходимо для проведения отдельной итерации. Контроль вычислений при первом алгоритме представляет, таким образом, достаточно трудоемкую работу. Нам неизвестны оценки, позволяющие указать число итераций, после которого рационально производить контроль. Исследование этого вопроса по крайней мере для отдельных классов задач представляет практический интерес.

6.3. Проиллюстрируем на примере порядок вычислений при использовании первого алгоритма метода последовательного улучшения плана.

Пример 1. Требуется обратить в максимум линейную форму

$$L(X) = 3x_1 - x_2 + 8x_3 + 2x_4 - x_5 + 9x_6$$

при условиях

$$-6x_1 + 9x_2 + 3x_3 - 2x_5 - x_6 \leq 12,$$

$$-4x_2 + 3x_3 - 3x_4 + x_5 - x_6 \leq 5,$$

$$2x_1 + 8x_2 - 5x_3 + 6x_4 - 8x_5 + 4x_6 \leq 20,$$

$$-x_1 - 3x_2 - 4x_3 - 8x_4 + 4x_6 \leq 10,$$

$$5x_1 + x_2 + 2x_3 + 4x_4 + 9x_5 + 5x_6 \leq 24.$$

$$x_j \geq 0, \quad j=1, 2, \dots, 6.$$

Чтобы привести задачу к канонической форме, введем пять дополнительных неотрицательных переменных x_7, \dots, x_{11} . Им соответствуют единичные векторы условий. Естественно выбрать единичные векторы A_7, \dots, A_{11} в качестве исходного базиса. Весь процесс решения задачи приведен в табл. 4.2. Таблица состоит из частей, отвечающих отдельным итерациям метода и пронумерованных в соответствии с номерами итераций.

Таблицу 0 (исключая ее верхнюю и последнюю строки) заполняют составляющие вектора ограничений (A_0) и векторов условий

Таблица 4.2 (0—7)

[illegible]

Продолжение табл. 4.2

\backslash	C	—	—	—	3	—1	8	2	—1	9							
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	0	N_T
	1		A_7	15,929	—5,821	9,821	1,857		—3,143	1			0,143	0,107		8,577	
	2		A_8	11,071	0,821	—0,821	1,643		—1,857		1		0,357	—0,107		6,739	
→	3	2	A_4	0,714	0,214	0,786	—0,071	1	—0,571				0,071	—0,071		—	
	4	9	A_6	3,929	0,179	0,821	—1,143		—1,143	1			0,143	0,107		—	
←	5		A_{11}	1,500	3,250	—6,250	8,000		17,000				—1,000	—0,250	1	0,188	
	6	—	—	36,786	—0,964	9,964	—18,429		—10,429				1,429	0,821		—	
	1		A_7	15,580	—6,576	11,272			—7,089	1			0,375	0,165	—0,232	1,382	
	2		A_8	10,736	0,154	0,462			—5,348		1		0,562	—0,056	—0,205	23,295	
←	3	2	A_4	0,728	0,243	0,730		1	—0,420				0,062	—0,074	0,009	0,997	
	4	9	A_6	4,143	0,643	—0,071			1,286	1				0,071	0,143	—	
→	5	8	A_3	0,188	0,406	—0,781	1		2,125				—0,125	—0,031	0,125	—	
	6	—	—	40,241	6,522	—4,433			28,732				—0,875	0,246	2,304	—	

Продолжение табл. 4.2

C	—	—	3	—1	8	2	—1	9								
N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	θ	N_T
1		A_7	4,343	-10,333			-15,443	-0,609		1			-0,590	1,303	-0,370	—
2		A_8	10,303				-0,633	-5,803			1		0,523	-0,009	-0,211	19,702
\rightarrow \leftarrow	3	-1 A_2	0,997	0,333	1		1,370	-0,575					0,086	-0,101	0,012	11,643
4	9	A_6	4,214	0,667			0,098	1,245	1				0,006	0,064	0,144	689,0
5	8	A_3	0,966	0,667		1	1,070	1,676					-0,058	-0,110	0,135	—
6	—	—	44,661	8,000			6,073	26,183					-0,495	-0,202	2,358	—
4																
1		A_7	11,214	-8,036	6,893		-6	-4,571		1			0,607	-0,286	18,471	
\leftarrow	2	A_8	4,214	-2,036	-6,107		-9	-1,571			1		0,607	-0,286	6,941	
\rightarrow	3	A_9	11,643	3,893	11,697		16	-6,714				1	-1,179	0,143	—	
4	9	A_6	4,143	0,643	-0,071			1,286	1				0,071	0,143	58,0	
5	8	A_3	1,643	0,893	0,679	1	2	1,286					-0,179	0,143	—	
6	—	—	50,429	9,929	5,786		14	22,857					-0,786	2,429	—	
5																

4

5

Продолжение табл. 4.2

	—	—	—	3	—1	8	2	—1	9										
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	θ	N_T		
← 1	1	A_7	7,000	—6,000	13,000	—	—	3,000	—3,000	1	—1,000	—	—	—	—	—	0,538		
→ 2	2	A_{10}	6,941	—3,353	—10,059	—	—	—14,824	—2,588	—	—	1,647	—	1	—0,471	—			
3	3	A_9	19,824	—0,059	—0,176	—	—	—1,471	—9,765	—	—	1,941	1	—	—0,412	—			
4	4	9 A_5	3,647	0,882	0,647	—	—	1,059	1,471	1	—0,118	—	—	—	0,176	5,636			
5	5	8 A_3	2,882	0,294	—1,118	—	1	—0,647	0,824	—	—	0,294	—	—	0,059	—			
6	6	—	—	55,882	7,294	—2,118	—	2,353	20,824	—	—	1,294	—	—	2,059	—			
→ 1	1	—1 A_2	0,538	—0,462	1	—	—	0,231	—0,231	0,077	—0,077	—	—	—	—	—			
2	2	A_{10}	12,357	—7,995	—	—	—	—12,502	—4,910	0,774	0,873	—	—	1	—0,471	—			
3	3	A_9	19,919	—0,140	—	—	—	—1,430	—9,805	0,014	1,928	—	1	—	—0,412	—			
4	4	9 A_8	3,299	1,181	—	—	—	0,910	1,620	1	—0,050	—0,068	—	—	0,176	—			
5	5	8 A_3	3,484	—0,223	—	1	—	—0,389	0,566	0,086	0,208	—	—	—	0,059	—			
6	6	—	—	57,023	0,632	—	—	2,842	20,335	0,163	1,131	—	—	—	2,059	—			

$(A_1, A_2, \dots, A_{11})$. Компоненты вектора ограничений входят в таблицу как базисные составляющие начального опорного плана, а компоненты векторов условий — как коэффициенты разложения векторов A_j по векторам единичного базиса.

Столбец S_X не заполняется, поскольку дополнительным переменным соответствуют нулевые коэффициенты линейной формы.

Последняя строка таблицы 0 содержит значения оценок Δ_j векторов условий. Параметры $\Delta_j = x_{m+1,j}$ вычисляются по формулам (6.2). В нашем случае $c_{s_j} = 0$. Поэтому в строку Δ_j переносятся элементы верхней строки (С) с обратным знаком. Значение линейной формы на начальном плане равно нулю.

Среди оценок Δ_j векторов условий имеются отрицательные величины. Следовательно, исходный опорный план не является оптимальным. В разложении каждого из векторов условий с отрицательными оценками содержатся положительные коэффициенты x_{ij} . Следовательно, нет оснований считать задачу неразрешимой. Мы имеем, таким образом, дело со случаем 3°.

Наименьшая оценка $\Delta_j = -9$ соответствует вектору условий A_6 . Вектор A_6 подлежит вводу в базис. Последние три элемента (x_{36} , x_{46} , x_{56}) направляющего столбца A_6 положительны. Поэтому значения θ вычисляются и записываются только для третьей, четвертой и пятой позиций столбца θ . Наименьший элемент последнего столбца соответствует четвертой позиции базиса и равен $\theta_0 = \frac{x_{40}}{x_{46}} = \frac{10}{4} = 2,5$. Это значит, что четвертая строка является направляющей строкой преобразования, и помещенный в ней вектор A_{10} подлежит исключению из базиса.

Переходим к составлению таблицы, отвечающей первой итерации. В столбце B_X в четвертой позиции базиса место вектора A_{10} занимает вектор условий A_3 . Соответствующий ему коэффициент линейной формы c_6 помещается в столбец S_X . Таблица 1 (столбцы A_0, A_1, \dots, A_{11}) заполняется по данным предыдущей таблицы (0) в соответствии с рекуррентными формулами

$$x'_{ij} = x_{ij} - \frac{x_{4j}}{x_{46}} x_{i6}, \quad i=0, 1, \dots, 6, \quad j=0, 1, \dots, 11, \quad (6.3)$$

$$x'_{4j} = \frac{x_{4j}}{x_{46}}, \quad j=0, 1, 2, \dots, 11. \quad (6.4)$$

Направляющий элемент (x_{rk}) равен $x_{46} = 4$. Следовательно, четвертая строка таблицы 1 получается из четвертой строки таблицы 0 делением на 4. Остальные элементы таблицы 1, в том числе составляющие опорного плана и оценки векторов условий, вычисляются по рекуррентным формулам (6.3). Например:

$$x'_{23} = x_{23} - \frac{x_{43}}{x_{46}} x_{26} = 3 - \frac{-4(-1)}{4} = 2,$$

$$x'_8 = x'_{20} = x_{20} - \frac{x_{40}}{x_{46}} x_{26} = 5 - \frac{10(-1)}{4} = 7,5,$$

$$\Delta'_2 = x'_{02} = x_{02} - \frac{x_{42}}{x_{46}} x_{06} = 1 - \frac{-3(-9)}{4} = -5,75.$$

Следующая итерация ведется по тем же правилам. Стрелками в каждой таблице отмечены векторы, введенные в базис на предыдущем шаге, и векторы, подлежащие исключению из базиса. В таблице 7 все значения оценок Δ_j неотрицательны. Это значит, что план $X^{(7)}$ является решением задачи.

Для контроля вычислений целесообразно на некоторых этапах расчета определять оценки Δ_j не только по рекуррентным формулам (6.1), но и непосредственно по формуле (6.2). Вычислим, например, значение Δ'_2 в таблице 1:

$$\begin{aligned}\Delta'_2 &= \sum_{i=1}^5 x'_{i2} c_{s_i} - c_2 = \\ &= 8,25 \cdot 0 - 4,75 \cdot 0 + 11 \cdot 0 - 0,75 \cdot 9 + 4,75 \cdot 0 - (-1) = -5,75.\end{aligned}$$

Тот же результат мы получили выше по рекуррентной формуле (6.3).

§ 7. ВТОРОЙ АЛГОРИТМ

7.1. Вторая форма признака оптимальности приводит к еще одной реализации метода последовательного улучшения плана, к так называемому второму алгоритму или методу обратной матрицы.

Метод последовательного улучшения плана в форме, близкой к излагаемой в настоящем параграфе, был впервые применен Л. В. Канторовичем к одной из частных задач линейного программирования [21, 19]. Позже (в 1951 г.) этот алгоритм как одна из реализаций так называемого метода разрешающих множителей был использован для решения общей задачи линейного программирования [22].

Будем рассматривать задачу линейного программирования в канонической форме. Пусть X — опорный план задачи с базисом $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Обозначим множество индексов векторов базиса (номера базисных переменных) через I_X .

Составим квадратную матрицу A_X порядка m . Столбцами A_X являются векторы условий базиса. Поэтому определитель матрицы A_X отличен от нуля и существует обратная матрица A_X^{-1} . Покажем, что с помощью матрицы A_X^{-1} и условий задачи можно составить достаточно компактную схему вычисления параметров, необходимых для последовательного улучшения плана.

Для реализации метода последовательного улучшения плана необходимо, имея начальный опорный план и его базис, уметь на каждом шаге выбирать вектор условий A_k , подлежащий включению в базис, и вектор A_{s_r} , подлежащий удалению из базиса. Вектор A_k — это вектор условий с отрицательной относительной оценкой Δ_k (обычно $\Delta_k = \min_j \Delta_j$). Вектор A_{s_r} — это вектор, на котором достигается минимум отношения x_{i0}/x_{ik} при $x_{ik} > 0$. Таким образом, для перехода от одного плана к другому с большим значением линейной формы необходимо уметь вычислять на каждом шаге базисные компоненты x_{i0} опорного плана, оценки Δ_j векторов условий относительно базиса опорного плана и коэффициенты x_{ik} разложения вектора A_k , подлежащего вводу в базис, по текущему базису. В первом алгоритме для определения перечисленных параметров на каждой итерации вычислялись коэффициенты разложения x_{ij} всех векторов условий A_j по векторам базиса. Как мы сейчас увидим, все параметры, необходимые для оценки плана и перехода к лучшему плану, можно получить, преобразовывая от шага к шагу лишь элементы матрицы A_X^{-1} .

Действительно, базисные составляющие опорного плана получаются как элементы произведения матрицы A_X^{-1} на вектор ограничений B .

Оценки Δ_j векторов условий можно, как видно из § 2, вычислять по формулам

$$\Delta_j = z_j - c_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j = (\Lambda, A_j) - c_j, \quad (7.1)$$

$$j = 1, 2, \dots, n,$$

где параметры λ_i удовлетворяют уравнениям

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j \quad \text{при} \quad j \in I_X,$$

или, что то же самое,

$$\Lambda A_X = C_X. \quad (7.2)$$

Здесь C_X — вектор-строка из коэффициентов линейной формы, отвечающих базисным переменным. Следовательно,

$$\Lambda = C_X A_X^{-1}. \quad (7.3)$$

Формулы (7.1)—(7.3) позволяют, если известны элементы обратной матрицы A_X^{-1} и условия задачи, вычислить оценки Δ_j векторов условий. Коэффициенты x_{ik} разложения вектора A_k по текущему базису вычисляются как элементы произведения матрицы A_X^{-1} на вектор условий A_k . Коэффициенты x_{ik} вместе с базисными составляющими опорного плана определяют вектор, подлежащий исключению из базиса. Кроме того, знаки коэффициентов x_{ik} позволяют судить о неразрешимости задачи. Естественно, что неразрешимость задачи, если она имеет место, при втором алгоритме обнаруживается, вообще говоря, позже, чем при первом алгоритме, где есть возможность оценить знаки x_{ij} при всех векторах A_j с отрицательными оценками.

Таким образом, для реализации метода последовательного улучшения плана достаточно уметь вычислять на каждом шаге матрицу A_X^{-1} , обратную матрицу, составленную из векторов текущего базиса.

Элементы столбцов обратной матрицы A_X^{-1} удобно рассматривать как коэффициенты e_{ij} разложения единичных векторов e_j ($j=1, 2, \dots, m$) по векторам базиса. В § 5 были получены рекуррентные формулы (5.21), связывающие параметры e_{ij} двух последовательных итераций. Напомним, что e_{ij} при $i=1, 2, \dots, m, j=1, 2, \dots, m$ — коэффициенты разложения единичных векторов по векторам базиса; $e_{i0} = x_{i0}$ — базисные переменные опорного плана; $e_{m+1,j} = \lambda_j$ — оценки условий задачи относительно данного плана; $e_{m+1,0} = \lambda_0 = L(X)$ — значение линейной формы.

Рекуррентные формулы (5.21) — основа второго алгоритма метода последовательного улучшения плана. Роль, которую играет в вычислительной схеме обратная матрица A_X^{-1} , оправдывает название «метод обратной матрицы», которое иногда присваивают второму алгоритму.

Результаты вычислений по второму алгоритму удобно сводить в основные таблицы (вида табл. 4.3) и вспомогательную таблицу (вида табл. 4.4); столбцы e_0, e_1, \dots, e_m основных таблиц (все $m+1$ позиции) обычно называют главной частью этих таблиц.

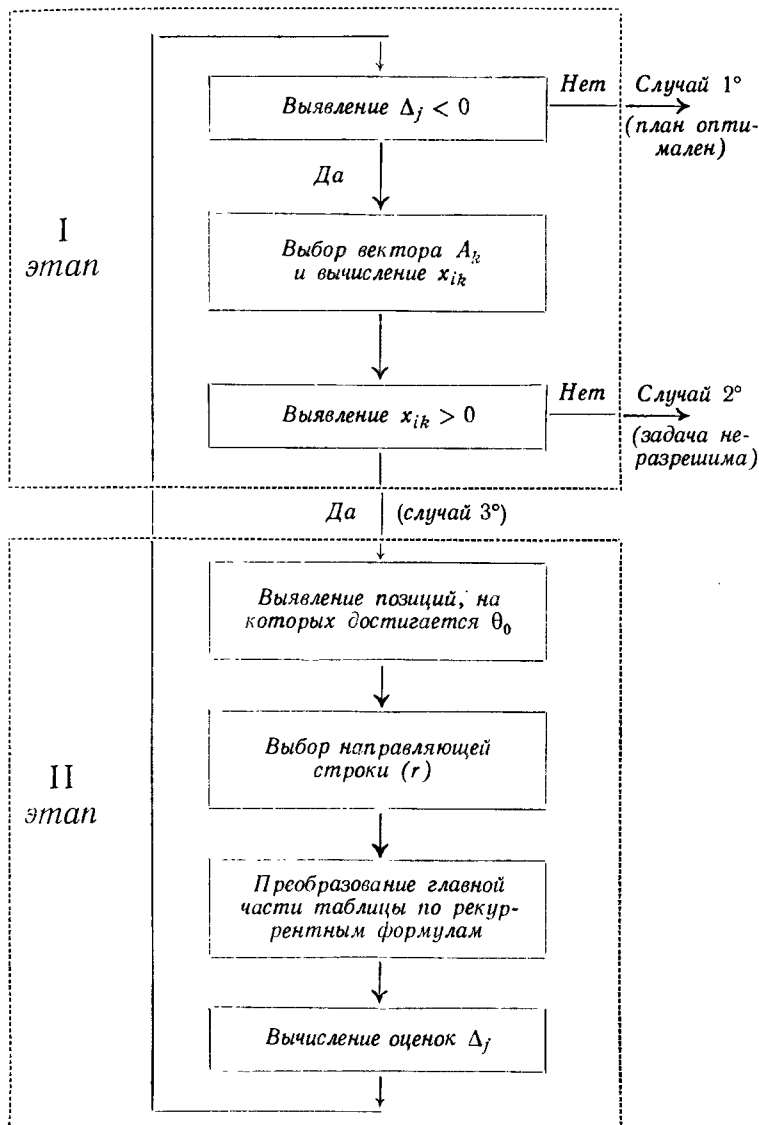


Рис. 4.2.

Таблица 4.3

N	C_X	B_X	e_0	e_1	e_2	...	e_m	A_k	θ
1	c_{s_1}	A_{s_1}	$e_{10}^{(l)}$	$e_{11}^{(l)}$	$e_{12}^{(l)}$...	$e_{1m}^{(l)}$	$x_{1k}^{(l)}$	
2	c_{s_2}	A_{s_2}	$e_{20}^{(l)}$	$e_{21}^{(l)}$	$e_{22}^{(l)}$...	$e_{2m}^{(l)}$	$x_{2k}^{(l)}$	
...
r	c_{s_r}	A_{s_r}	$e_{r0}^{(l)}$	$e_{r1}^{(l)}$	$e_{r2}^{(l)}$...	$e_{rm}^{(l)}$	$x_{rk}^{(l)}$	θ_0
...
m	c_{s_m}	A_{s_m}	$e_{m0}^{(l)}$	$e_{m1}^{(l)}$	$e_{m2}^{(l)}$...	$e_{mm}^{(l)}$	$x_{mk}^{(l)}$	
$m+1$	—	—	$L^{(l)}$	$\lambda_1^{(l)}$	$\lambda_2^{(l)}$...	$\lambda_m^{(l)}$	$\Delta_k^{(l)}$	—

На рис. 4.2 изображена блок-схема отдельной итерации второго алгоритма метода последовательного улучшения плана.*

Для контроля вычислений при втором алгоритме естественно использовать две возможности для определения величин λ_j . В любой итерации параметры λ_j могут быть вычислены непосредственно (по формуле (7.3)) или по рекуррентным формулам (5.21).

7.2. Оценим трудоемкость вычислений в отдельной итерации второго алгоритма метода последовательного улучшения плана. Мы уже отмечали, что трудоемкость вычислений целесообразно оценивать по количеству делений и умножений, необходимых для проведения каждой итерации.

Во втором алгоритме операция деления встречается в каждой итерации дважды: при вычислении строки но-

Т а б л и ц а 4.4

N	B	A_1	A_2	...	A_k	...	A_n
1	b_1	a_{11}	a_{12}	...	a_{1k}	...	a_{1n}
2	b_2	a_{21}	a_{22}	...	a_{2k}	...	a_{2n}
...
m	b_m	a_{m1}	a_{m2}	...	a_{mk}	...	a_{mn}
$m+1$	C	c_1	c_2	...	c_k	...	c_n
0	Δ	Δ_1	Δ_2	...	Δ_k	...	Δ_n
1	Δ'	Δ'_1	Δ'_2	...	Δ'_k	...	Δ'_n
2	Δ''	Δ''_1	Δ''_2	...	Δ''_k	...	Δ''_n
...

вой основной таблицы, отвечающей вновь введенному в базис вектору, и при вычислении столбца θ . В первом случае число делений равно $m+1$, а во втором не превышает m . Операция умножения встречается во втором алгоритме трижды: при преобразовании основной таблицы по рекуррентным формулам, при вычислении оценок Δ_j и при определении коэффициентов разложения x_{ik} вектора A_k , вводимого в базис, по векторам базиса. Для преобразования главной части основной таблицы требуется $m(m+1)$ умножений. Для вычисления оценки Δ_j каждого вектора условий A_j необходимо m умножений. Всего подлежат вычислению $n-m$ оценок векторов условий (по

числу внебазисных переменных: оценки векторов базиса не вычисляются: они равны нулю). Следовательно, для вычисления всех оценок необходимо при каждой итерации произвести $m(n-m)$ умножений. Наконец, для вычисления всех x_{ik} ($i=1, 2, \dots, m$) по формулам

$$x_{ik}^{(l)} = \sum_{j=1}^m e_{ij}^{(l)} a_{jk} \text{ требуется еще } m^2 \text{ умножений. Таким образом,}$$

проведение каждой итерации во втором алгоритме связано с числом делений, не превышающим $2m+1$, и числом умножений, равным

$$m(m+1) + m(n-m) + m^2 = m(n+m+1).$$

Для контроля расчетов целесообразно через определенное количество шагов вычислять параметры λ не только по рекуррентным формулам, но и непосредственно по формулам (7.3). Каждый этап контроля требует, таким образом, m^2 дополнительных операций умножения.

Сравнивая трудоемкость отдельных этапов каждой итерации, необходимо сделать следующие замечания.

При $n \gg m$ вычисление оценок становится наиболее трудоемкой операцией во втором алгоритме. Вычисление всех оценок производится для того, чтобы сравнить Δ_j между собой и выбрать наименьшее из них. Вектор A_k с оценкой $\Delta_k = \min_j \Delta_j$ вводится в базис. По-видимому,

общее количество операций существенно сократится, если вычислять оценки Δ_j до первой отрицательной величины и вводить в базис вектор, отвечающий этой оценке. Число шагов при этом может увеличиться. Однако возможное увеличение числа итераций, как правило, окупается уменьшением трудоемкости вычислений в каждой итерации. В каждой новой итерации порядок, в котором вычисляются оценки, вообще говоря, произволен. Интуитивно, однако, представляется наиболее целесообразным следующий порядок. Для исходного опорного плана вычисляются оценки $\Delta_1, \dots, \Delta_{p_1}$, где Δ_{p_1} — первая отрицательная оценка. Для следующего опорного плана вычисляются оценки $\Delta'_{p_1+1}, \dots, \Delta'_{p_2}$, где Δ'_{p_2} — первое отрицательное число в последовательности $\Delta'_{p_1+1}, \Delta'_{p_1+2}, \dots$. Если все $\Delta'_q \geq 0$ при $q = p_1 + 1, p_1 + 2, \dots, n$, следует перейти к вычислению $\Delta'_1, \Delta'_2, \dots$.

7.3. Узким местом современных вычислительных машин является оперативная память. Ниже будет намечена модификация второго алгоритма, позволяющая экономнее использовать память машины [15].

В приведенной вычислительной схеме необходимо в каждой итерации запоминать всю обратную матрицу $A_X^{-1} = \|e_{ij}\|$. В так называемой мультипликативной форме второго алгоритма регистрации подлежит значительно меньшее число данных.

Мультипликативная форма основана на следующих соображениях. Пусть X и X' — два последовательных опорных плана задачи. Обозначим соответствующие матрицы векторов базиса через A_X и $A_{X'}$:

$$A_X = (A_{s_1}, \dots, A_{s_r}, \dots, A_{s_m}), \quad A_{X'} = (A_{s_1}, \dots, A_{k_1}, \dots, A_{s_m}).$$

Как легко непосредственно проверить, обратные матрицы связаны соотношениями

$$A_{X'}^{-1} = E^r A_X^{-1}, \quad (7.4)$$

где

$$E^r = \begin{pmatrix} 1 & 0 & \dots & y_{1k} & \dots & 0 \\ 0 & 1 & \dots & y_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & y_{rk} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & y_{mk} & \dots & 1 \end{pmatrix},$$

$$y_{ik} = -\frac{x_{ik}}{x_{rk}}, \quad i = 1, 2, \dots, m, \quad i \neq r, \quad y_{rk} = \frac{1}{x_{rk}}.$$

Соотношение (7.4) эквивалентно применению рекуррентных формул (5.21) при $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$.

Обычно решение задачи линейного программирования начинается с единичного базиса. Ему соответствует единичная матрица E . Поэтому после первой итерации, когда в базис вместо вектора $A_{s_{r_1}}$ вводится вектор A_{k_1} , обратная матрица базиса полученного опорного плана X' может вычисляться по формуле

$$A_{X'}^{-1} = E^r E,$$

а после l -й итерации

$$A_{X^{(l)}}^{-1} = E^l E^{l-1} \dots E^1 E. \quad (7.5)$$

Матрица E^l определяется $m+1$ числом $(r, y_{1k}, \dots, y_{mk})$. Следовательно, при $l < m$ запись обратной матрицы в форме (7.5) меньше загружает память машины, чем обычная запись, требующая запоминания при каждой итерации m^2 чисел e_{ij} .

Часто для задач со слабо заполненными матрицами условий большинство чисел y_{ik} ($i = 1, 2, \dots, m$) оказываются равными нулю. В таких случаях столбцы $(r, y_{1k}, \dots, y_{mk})$ хранят в памяти машины в экономной форме (запоминаются только ненулевые элементы и их позиции).

Как уже отмечалось, опыт показывает, что число итераций, необходимых для решения задачи линейного программирования, обычно представляет собой величину порядка m . Поэтому следует ожидать, что мультипликативная форма почти всегда будет компактнее, чем обычная запись второго алгоритма. В тех случаях, когда число итераций превысит m и по тем или иным причинам нецелесообразно преобразовывать обратную матрицу базиса опорного плана $X^{(m)}$ к единичной*), следует, начиная с $(m+1)$ -й итерации, перейти к обычной форме регистрации обратной матрицы.

Наметим схему вычислений в отдельной итерации мультипликативной формы второго алгоритма. Пусть проведено l итераций, в результате которых установлено, что $X^{(l)}$ не является оптимальным планом и нет оснований говорить о неразрешимости задачи. Определение вектора, подлежащего вводу в базис, и вектора, подлежащего исключению из базиса, производится, естественно, по общим правилам метода последовательного улучшения плана. Компоненты очередного опорного плана вычис-

*) Преобразование обратной матрицы A_X^{-1} к единичной на каком-либо шаге процесса решения задачи соответствует переходу к эквивалентной задаче, векторы условий и вектор ограничений которой получены из векторов условий и вектора ограничений исходной задачи умножением слева на A_X^{-1} .

ляются, как и в обычной форме второго алгоритма, по рекуррентным формулам (5.21).

Вектор оценок условий задачи (Λ) в $(l+1)$ -й итерации вычисляется как произведение

$$\Lambda_{l+1} = C_X E^{r_l} E^{r_{l-1}} \dots E^{r_1} E.$$

Вектор оценок Λ векторов условий определяется, как обычно во втором алгоритме, по формуле

$$\Lambda = \Lambda A_X - C.$$

Наконец, коэффициенты x_{ik} разложения вектора A_k , подлежащего включению в базис, по векторам базиса, определяются из соотношения

$$A_k^{(l+1)} = E^{r_l} E^{r_{l-1}} \dots E^{r_1} A_k.$$

Здесь

$$A_k^{(l+1)} = (x_{1k}^{(l+1)}, \dots, x_{mk}^{(l+1)}), \quad A_k = A_k^{(0)} = (a_{1k}, \dots, a_{mk}).$$

Как видим, мультипликативная форма второго алгоритма позволяет не только экономно использовать память машины, но и в ряде случаев, по крайней мере в начальных итерациях, приводит к существенно менее трудоемким вычислениям.

7.4. Проиллюстрируем порядок решения задачи вторым алгоритмом метода последовательного улучшения плана на примере, рассмотренном в § 6 в связи с первым алгоритмом.

В основной табл. 4.5 и вспомогательной табл. 4.6 записан процесс решения примера 1 (см. § 6). Табл. 4.5 разбита на 8 частей (8 основных таблиц). Главная часть основной таблицы 0 содержит исходные данные для первой итерации. Последние два столбца и последняя строка основной и вспомогательной таблиц вместе с главной частью последующей основной таблицы 1 соответствуют очередной итерации алгоритма. В верхней части вспомогательной таблицы перечислены компоненты вектора ограничений и всех векторов условий. В $(m+1)$ -й (шестой) строке указаны соответствующие значения коэффициентов линейной формы. Строки $\Delta^{(l)}$ для каждой итерации заполняются по формулам (7.1). Исходный базис составлен из единичных векторов, отвечающих дополнительным переменным x_7, x_8, \dots, x_{11} . Соответствующие коэффициенты линейной формы равны нулю. Поэтому в основной таблице 0 все $\lambda_j = 0$ и в строке 0 вспомогательной таблицы $\Delta_j = -c_j$. Вектор A_6 с наименьшей оценкой ($\Delta_6 = -9$) подлежит вводу в базис.

Таблица 4.5 (0—7)

N	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	N_T
1		A_7	12	1					-1	—	0
2		A_8	5		1				-1	—	
3		A_9	20			1			4	5	
← 4		A_{10}	10				1		4	2,5	
5		A_{11}	24					1	5	4,8	
6	—	—							-9	—	
1		A_7	14,5	1			0,25		-2	—	1
2		A_8	7,5		1		0,25		-5	—	
← 3		A_9	10,0			1	-1,00		14	0,714	
→ 4	9	A_8	2,5				0,25		-2	—	
5		A_{11}	11,5				-1,25	1	14	0,822	
6	—	—	22,5				2,25		-20	—	

Продолжение табл. 4.5

N	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	N_T
2											
1		A_7	15,929	1		0,143	0,107		1,857	8,577	
2		A_8	11,071		1	0,357	-0,107		1,643	6,799	
→	3	A_4	0,714			0,071	-0,071		-0,071	—	
	4	A_6	3,929			0,143	0,107		-1,143	—	
←	5	A_{11}	1,500			-1,000	-0,250	1	8,000	0,188	
	6	—	36,786			1,429	0,821		-18,429	—	
3											
1		A_7	15,580	1		0,375	0,165	-0,232	11,272	1,382	
2		A_8	10,763		1	0,562	-0,056	-0,205	0,462	23,295	
←	3	A_4	0,728			0,062	-0,074	0,009	0,730	0,997	
	4	A_6	4,143				0,071	0,143	-0,071	—	
→	5	A_3	0,188			-0,125	-0,031	0,125	-0,781	—	
	6	—	40,241			-0,875	0,246	2,304	-4,433	—	

Продолжение табл. 4.5

	N	CX	BX	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	N_T
	1		A_7	4,343	1		-0,590	1,303	-0,370	-0,590	—	
	2		A_9	10,303		1	0,523	-0,009	-0,211	0,523	19,702	
\leftrightarrow	3	-1	A_2	0,997			0,086	-0,101	0,012	0,086	11,643	4
	4	9	A_8	4,214			0,006	0,064	0,144	0,006	689,0	
	5	8	A_3	0,966			-0,058	-0,110	0,135	-0,058	—	
	6	—	—	44,661			-0,495	-0,202	2,358	-0,495	—	
	1		A_7	11,214	1			0,607	-0,286	0,607	18,471	
\leftarrow	2		A_8	4,214		1		0,607	-0,286	0,607	6,941	
\rightarrow	3		A_9	11,643			1	-1,179	0,143	-1,179	—	
	4	9	A_6	4,143				0,071	0,143	0,071	58,00	
	5	8	A_3	1,643				-0,179	0,143	-0,179	—	
	6	—	—	50,429				-0,786	2,429	-0,786	—	

Продолжение табл. 4.5

	N	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	N_T
\leftarrow	1		A_7	7,000	1	-1,000				13,000	0,538	
\rightarrow	2		A_{10}	6,941		1,647		1	-0,471	-10,059	-	
	3		A_9	19,824		1,941	1		-0,412	-0,176	-	
	4	9	A_6	3,647		-0,118			0,176	0,647	5,636	
	5	8	A_3	2,882		0,294			0,059	-1,118	-	
	6	-	-	55,882		1,294			2,059	-2,118	-	
\rightarrow	1	-1	A_2	0,538	0,077	-0,077						
	2		A_{10}	12,357	0,774	0,873		1	-0,471			
	3		A_9	19,919	0,014	1,928	1		-0,412			
	4	9	A_6	3,299	-0,050	-0,068			0,176			
	5	8	A_3	3,484	0,086	0,208			0,059			
	6	-	-	57,023	0,163	1,131			2,059		-	

Таблица 4.6

N	B	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁
1	12	-6	9	3		-2	-1	1				
2	5		-4	3	-3	1	-1		1			
3	20	2	8	-5	6	-8	4			1		
4	10	-1	-3	-4	-8		4				1	
5	24	5	1	2	4	9	5					1
6	$\frac{C}{\Delta}$	3	-1	8	2	-1	9					
0	Δ	-3	1	-8	-2	1	$\boxed{-9}$					
1	Δ^I	-5,25	-5,75	-17	$\boxed{-20}$	1					2,25	
2	Δ^{II}	-0,964	9,964	$\boxed{-18,429}$		-10,429				1,429	0,821	
3	Δ^{III}	6,522	$\boxed{-4,433}$			28,732				-0,875	0,246	2,304
4	Δ^{IV}	8,000			6,073	26,183				$\boxed{-0,495}$	-0,202	2,358
5	Δ^V	9,929	5,786		14,000	22,857					$\boxed{-0,786}$	2,429
6	Δ^{VI}	7,294	$\boxed{-2,118}$		2,353	20,824			1,294			2,059
7	Δ^{VII}	0,632			2,842	20,335		0,163	1,131			2,059

Столбец A_k основной таблицы 0 заполняется компонентами x_{i6} разложения A_6 по векторам базиса. Поскольку начальный базис составлен из единичных векторов, $x_{i6} = a_{i6}$. В $(m+1)$ -ю позицию столбца A_k помещается оценка Δ_6 вектора A_6 . В столбце A_6 имеются положительные составляющие. Следовательно, мы имеем дело со случаем 3°. Переходим к определению вектора, подлежащего исключению из базиса. В столбце θ заполняются позиции, для которых $x_{ik} > 0$ (позиции 3, 4 и 5). Наименьшее значение $\theta = \theta_0 = 2,5$ соответствует вектору базиса A_{10} , расположенному в четвертой позиции базиса. Четвертая строка является, таким образом, направляющей строкой преобразования, и вектор A_{10} базиса должен быть заменен вектором A_6 . Направляющим элементом (x_{rk}) преобразования является компонента $x_{46} = 4$, находящаяся на пересечении строки 4 и столбца A_6 .

Главная часть основной таблицы 1 вычисляется по элементам главной части таблицы 0 в соответствии с рекуррентными формулами. Элементы строки 4 таблицы 1 равны соответствующим элементам четвертой строки таблицы 0, разделенным на направляющий элемент $x_{46} = 4$. Все остальные элементы главной части таблицы (в том числе и составляющие столбца e_0 и строки λ) вычисляются по формуле

$$e_{ij}^{(l+1)} = e_{ij}^{(l)} - \frac{e_{rj}^{(l)}}{x_{rk}^{(l)}} x_{ik}^{(l)} = e_{ij}^{(l)} - e_{rj}^{(l+1)} x_{ik}^{(i)},$$

$$i = 1, 2, \dots, m+1, \quad j = 0, 1, 2, \dots, m.$$

Например:

$$x'_{30} = e'_{30} = e_{30} - \frac{e_{40}}{x_{46}} x_{36} = 20 - \frac{10.4}{4} = 10,$$

$$e'_{54} = e_{54} - \frac{e_{44}}{x_{46}} x_{56} = 0 - \frac{1.5}{4} = -1,25,$$

$$L(X') = e'_{m+1,0} = \lambda'_0 = e'_{60} = e_{60} - \frac{e_{40}}{x_{46}} x_{66} = 0 - \frac{10(-9)}{4} = 22,5.$$

Здесь $x_{66} = \Delta_6$ — оценка вектора A_6 , введенного в базис.

Теперь следует вычислить по формуле (7.1) строку Δ вспомогательной табл. 4.6.

Имеем, например,

$$\Delta'_4 = (0.0 - 3.0 + 6.0 - 8.2,25 + 4.0) - 2 = -20.$$

Δ'_4 — наименьший элемент строки Δ' . В очередной базис вводится A_4 . Следующие итерации проводятся по тем же правилам. Для контроля следует в отдельных итерациях вычислять оценки условий λ_i не только по рекуррентным формулам, но и непосредственно по формуле (7.3).

§ 8. СРАВНИТЕЛЬНАЯ ОЦЕНКА АЛГОРИТМОВ

Расчеты по первому алгоритму связаны только с одним типом таблиц и более единообразны, чем вычисления по второму алгоритму. При ручном счете—это важное достоинство первого алгоритма.

Первый алгоритм позволяет быстрее обнаружить неразрешимость задачи, если она имеет место. Как мы увидим из § 9, первый алгоритм предпочтительней использовать при решении вспомогательной задачи, в которой определяется начальный опорный план исходной задачи. Наконец, первый алгоритм обеспечивает возможность включения в базис вектора A_k , при котором достигается максимально возможное увеличение линейной формы за итерацию $(-\Delta_k \theta_0^{(k)} = \max_j (-\Delta_j \theta_0^{(j)}))$.

Вычисления по второму алгоритму предъявляют менее жесткие требования к емкости оперативной памяти, чем расчеты по первому алгоритму. При первом алгоритме запоминается таблица, отвечающая всем векторам условий. Объем информации, запоминаемой при втором алгоритме, определяется только векторами базиса. В мультипликативной форме второго алгоритма на каждом шаге регистрируется еще меньше данных.

В практических задачах процент нулей в матрице условий обычно весьма велик. В первом алгоритме после элементарного преобразования нулевые компоненты векторов условий превращаются, вообще говоря, в ненулевые. Во втором алгоритме, в котором вся матрица условий не преобразовывается от таблицы к таблице, преимущество векторов условий с большим количеством нулей сохраняется на всем протяжении вычислений. В частности, во втором алгоритме весьма просто вычисляются оценки единичных векторов условий

$$\Delta_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j = \sum_{i=1}^m \delta_i (\lambda_i - c_j) = \lambda_j - c_j.$$

При решении задачи по второму алгоритму вместе с оптимальным планом исходной задачи получается оптимальный план двойственной задачи (вектор $\Lambda^* = (\lambda_1^*, \dots, \lambda_m^*)$, расположенный в $(m+1)$ -й строке последней основной таб-

лицы) и обратная матрица оптимального базиса. Эта особенность второго алгоритма весьма полезна для ряда приложений.

При решении задач вторым алгоритмом приходится чаще обращаться к исходным данным, чем при использовании первого алгоритма. Поэтому естественно, что накопление ошибок вычислений во втором алгоритме происходит медленнее, чем в первом.

Сравним теперь оба алгоритма по числу операций умножения и деления на каждую итерацию. В п. 6.2 было показано, что при решении задачи линейного программирования по первому алгоритму в каждой итерации производится не более $n+1$ делений и $m(n-m+1)$ умножений. В п. 7.2 было установлено, что каждая итерация при втором алгоритме метода требует не более $2m+1$ операций деления и $m(n+m+1)$ умножений. Как видим, второй алгоритм, связанный с преобразованием обратной матрицы и выбором вектора, вводимого в базис по наименьшей оценке Δ_j , требует, вообще говоря, большего числа операций, чем первый алгоритм.

Число умножений в каждой итерации второго алгоритма сокращается, если, как это указано в п. 7.3, не связывать выбор вектора, подлежащего включению в базис, с вычислением оценок Δ_j всех векторов условий. Однако в этом случае нельзя сравнивать трудоемкость обоих алгоритмов по числу операций на итерацию, поскольку решение задачи по разным алгоритмам потребует, вообще говоря, различного числа итераций.

В ряде случаев известное сокращение трудоемкости вычислений обеспечивает мультипликативная форма второго алгоритма.

Заметим, что сравнительная оценка трудоемкости первого и второго алгоритмов в статье Вагнера [5], на которую ссылаются в ряде работ, не может быть признана объективной. В этой статье указывается, что при $n > 3m$ первый алгоритм всегда связан с более трудоемкими вычислениями, чем второй. Определяя число операций умножения, используемых в первом алгоритме, автор предполагает, что оценки Δ_j вычисляются при первом алгоритме не по рекуррентным соотношениям (6.1), а непосредственно по формуле (7.1).

Вывод, полученный Вагнером, был бы справедлив, если бы было признано рациональным производить контроль вычислений после каждой итерации. В приведенной ниже таблице указано количество умножений, необходимое для проведения отдельной итерации и для контроля в первом и втором алгоритмах метода последовательного улучшения плана.

Таблица 4.7

	Первый алгоритм	Второй алгоритм
Проведение итерации . .	$m(n - m + 1)$	$m(n + m + 1)$
Контроль . . .	$m(n + 1)$	$m(m + 1)$
Итого . . .	$m(2n - m + 2)$	$m(2m + n + 2)$

Из таблицы видно, что при $n > 3m$ первый алгоритм оказывается более трудоемким, чем второй. Однако, как уже отмечалось, вряд ли целесообразно контролировать вычисления на каждом шаге метода.

Приведенная сравнительная оценка трудоемкости алгоритмов не учитывает специфических особенностей отдельных классов задач линейного программирования. В задачах с матрицами условий, в которых высок процент нулей (на практике встречается большое количество таких задач), число операций на один шаг метода во втором алгоритме меньше, чем в первом. Мы уже видели, что наиболее трудоемкий этап вычислений при втором алгоритме — вычисление оценок Δ_j — существенно сокращается для векторов условий, содержащих большое число нулей.

Пусть каждый столбец матрицы условий содержит не более am отличных от нуля элементов. Вычисление оценок Δ_j требует при этом $am(n - m)$ умножений. Для преобразования главной части основной таблицы необходимо, как и прежде, $m(m + 1)$ умножение, а для вычисления элементов направляющего столбца $A_k^{(b)} — m^2$ умножений. Общее количество операций умножения на каждом шаге второго алгоритма не превышает, таким

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (9.3)$$

Здесь можно считать все $b_i \geq 0$. В противном случае следовало бы в соответствующей строке поменять в обеих частях равенства все знаки на обратные.

Рассмотрим наряду со сформулированной задачей следующую вспомогательную задачу. Требуется обратить в максимум линейную форму

$$\tilde{L} = -(x_{n+1} + x_{n+2} + \dots + x_{n+m}) \quad (9.4)$$

при условиях

[illegible]

$$x_j \geq 0, \quad j = 1, 2, \dots, n + m. \quad (9.6)$$

Решим вспомогательную задачу методом последовательного улучшения плана. Начальный план вспомогательной задачи может быть непосредственно указан. Он составляется из компонент

$$\begin{aligned} x_j &= 0 & \text{при } j \leq n, \\ x_{n+i} &= b_i & \text{при } i = 1, 2, \dots, m. \end{aligned}$$

Нетрудно видеть, что этот план является опорным. Действительно, определитель, составленный из векторов условий при дополнительных переменных, равен единице. Следовательно, соответствующие векторы условий линейно независимы. Линейная форма \bar{L} ограничена сверху на множестве своих планов ($\bar{L} \leq 0$). Поэтому процесс последовательного улучшения плана через конечное число шагов приведет к оптимальному опорному плану вспомогательной задачи. Возможны два случая:

1°. Оптимальное значение \tilde{L} равно нулю.

2°. Оптимальное значение \bar{L} отрицательно.

В случае 1° оптимальный план вспомогательной задачи оказывается опорным планом исходной задачи. Действительно, при $\tilde{L}=0$ все $x_{n+i}=0$ ($i=1, 2, \dots, m$). Следовательно, решение вспомогательной задачи удовлетворяет условиям исходной задачи. Оптимальный план вспомогательной задачи совпадает, таким образом, с некоторым

планом исходной задачи. По построению этот план является опорным.

В случае 2° исходная задача не имеет ни одного плана, т. е. система условий исходной задачи несовместна. В этом нетрудно убедиться, если допустить противное, т. е. принять, что исходная задача имеет хотя бы один план. При таком допущении вспомогательная задача будет иметь план, у которого первые n составляющих совпадают с соответствующими компонентами исходного плана, а последующие m компонент равны нулю. Значение линейной формы для этого плана равно нулю, что противоречит условию, согласно которому максимальное значение \bar{L} отрицательно.

Итак, начальный опорный план любой задачи линейного программирования может быть вычислен в результате решения вспомогательной задачи с очевидным начальным опорным планом.

9.2. Далеко не во всех случаях имеет смысл разделять решение задачи линейного программирования на два этапа — вычисление начального опорного плана и определение оптимального плана. Рассмотрим прием, позволяющий объединить оба этапа. Сущность приема заключается в том, что вместо исходной задачи линейного программирования решается расширенная задача, имеющая другие опорные планы (один из них всегда легко указать), но те же решения, т. е. те же оптимальные планы, что и исходная задача. Множество опорных планов расширенной задачи содержит в качестве подмножества все опорные планы исходной задачи.

Рассмотрим наряду с исходной задачей линейного программирования следующую расширенную задачу. Требуется обратить в максимум линейную форму

$$\bar{L} = c_1 x_1 + \dots + c_n x_n - M(x_{n+1} + \dots + x_{n+m}),$$

где $M > 0$ достаточно большое число, а $X = (x_1, \dots, x_{n+m})$ удовлетворяет условиям (9.5), (9.6). Назовем расширенную задачу *M-задачей*.

Прием, излагаемый ниже, основан на следующих трех утверждениях, доказательство которых предоставляется читателю:

I. Если в оптимальном плане \tilde{X} M -задачи $x_{n+i}=0$ ($i=1, 2, \dots, m$), т. е. если $\tilde{X}=(x_1, \dots, x_n, 0, \dots, 0)$, то план $X=(x_1, x_2, \dots, x_n)$ является решением исходной задачи.

II. Всегда можно указать такое число $M_0 > 0$, что для любого $M > M_0$ из существования хотя бы одного плана исходной задачи вытекает соотношение

$$x_{n+i}=0 \quad (i=1, 2, \dots, m)$$

для оптимального плана M -задачи (если последняя разрешима).

III. Существует такое число M_1 , что при $M > M_1$ из разрешимости исходной задачи вытекает разрешимость связанной с ней M -задачи.

Приведенные положения обеспечивают возможность решения задачи линейного программирования с неизвестным заранее начальным опорным планом. Исходная задача сводится к M -задаче, для которой выбор начального опорного плана не представляет труда. Такой метод решения задачи линейного программирования называется M -методом. Он весьма эффективен при $m \ll n$.

Для решения задачи нет необходимости в вычислении величины M_0 . При определении значений Δ_j в процессе решения M -задачи следует полагать M больше любого сравниваемого с ним числа. Если в процессе решения M -задачи будет получен оптимальный план, для которого $x_{n+i}=0$ для $i=1, 2, \dots, n$, то первые n его компонент определяют решение исходной задачи. Если по крайней мере одна из составляющих x_{n+i} положительна при любом сколь угодно большом M , исходная задача не имеет ни одного плана, т. е. условия задачи не совместны. Если в процессе анализа M -задачи выявится ее неразрешимость, то это указывает на неразрешимость исходной задачи. Итак, M -метод позволяет вычислить оптимальный план любой разрешимой задачи линейного программирования без предварительного определения начального опорного плана.

9.3. Идея введения дополнительных переменных полезна всегда, когда ограничения искомых переменных

заданы неравенствами

$$A_1x_1 + A_2x_2 + \dots + A_nx_n \leq B.$$

(Составляющие вектора ограничения B , как и прежде, предполагаются неотрицательными.) В этом случае задача сводится к канонической форме, если ввести m дополнительных неотрицательных переменных x_{n+1}, \dots, x_{n+m} и соответствующие векторы условий A_{n+1}, \dots, A_{n+m} . Все компоненты вектора A_{n+k} , за исключением k -й, равны нулю; k -я составляющая A_{n+k} равна 1. В линейной форме коэффициенты при дополнительных переменных равны нулю, и, следовательно, величины x_{n+k} не оказывают влияния на значение формы.

Условия задачи, записанные в канонической форме, имеют вид

$$A_1x_1 + \dots + A_nx_n + A_{n+1}x_{n+1} + \dots + A_{n+m}x_{n+m} = B.$$

Матрица

$$(A_{n+1}, \dots, A_{n+m}) = \begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

— единичная матрица. Следовательно, дополнительные векторы образуют базис, которому соответствует начальный опорный план задачи

$$X = (\underbrace{0, 0, \dots, 0}_n, b_1, b_2, \dots, b_m).$$

Разложение векторов условий A_j по векторам начального единичного базиса не представляет труда. При единичном начальном базисе отпадает необходимость в решении системы уравнений для вычисления x_{ij} .

Отсутствие необходимости в решении систем линейных уравнений существенно упрощает вычисления в методе последовательного улучшения плана. Однако при этом следует иметь в виду, что, выбирая дополнительные векторы в качестве векторов начального базиса, мы исходим обычно из худшего первого приближения, чем в тех случаях, когда в базис с самого начала вводятся

векторы, соответствующие основным переменным. Количество приближений, требуемых для вычисления оптимального плана, при этом, как правило, увеличивается. Тем не менее, при использовании универсальных вычислительных машин введение базиса, соответствующего дополнительным переменным, оправдывает себя. Увеличение количества вычислений, которое может при этом произойти, связано только с дополнительными однообразными весьма простыми операциями.

9.4. Приведем некоторые замечания, позволяющие в ряде случаев упростить вычисление исходного опорного плана.

До сих пор мы связывали вспомогательную задачу и M -задачу с полным искусственным базисом — с m дополнительными единичными векторами. Трудоемкость вычислений заметно сокращается, если исходная задача содержит некоторое количество единичных векторов условий. В ряде случаев несложные преобразования позволяют уменьшить число искусственных переменных вспомогательной задачи (или M -задачи).

Пусть условия задачи записаны в таком виде:

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

где $b_i \geq 0$. Как уже указывалось, дополнительные неотрицательные переменные x_{n+i} ($i = 1, \dots, m$) позволяют свести задачу к каноническому виду. При этом значения дополнительных переменных $x_{n+i} = b_i$ определяют исходный опорный план задачи.

Пусть теперь условия задачи имеют вид

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

где $b_i \geq 0$. Дополнительные переменные x_{n+i} ($i = 1, 2, \dots, m$) позволяют и в этом случае свести задачу к канонической форме. Однако для получения исходного опорного плана здесь необходимо решить вспомогательную задачу (или

M -задачу) с одним искусственным вектором. Имеем

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

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

Пусть $b_s = \max_i b_i$. Преобразуем систему условий задачи:

$$\sum_{j=1}^n (a_{sj} - a_{ij})x_j - x_{n+s} + x_{n+i} = b_s - b_i, \quad i \neq s,$$

$$\sum_{j=1}^n a_{sj}x_j - x_{n+s} = b_s, \quad x_j \geq 0, \quad j = 1, 2, \dots, n+m.$$

Мы получили эквивалентную задачу, в которой при неотрицательных компонентах вектора ограничений имеется $m-1$ положительных единичных векторов условий. Определение исходного опорного плана связано, таким образом, лишь с одним искусственным вектором.

Пусть, наконец, условия задачи записаны в канонической форме:

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

Допустим, что все $b_i \geq 0$ и по крайней мере одна из компонент вектора ограничений положительна. Могут встретиться три случая:

1°. Среди векторов условий нет единичных векторов. В этом случае M -задача (или вспомогательная задача) содержит m искусственных переменных.

2°. Матрица $\|a_{ij}\|$ содержит r различных единичных векторов условий. В этом случае M -задача содержит $m-r$ искусственных векторов.

3°. Матрица условий $\|a_{ij}\|$ содержит единичную подматрицу порядка m . Здесь начальный план очевиден, и необходимость в M -методе или в решении вспомогательной задачи отпадает.

В ряде задач линейного программирования имеются те или иные основания выделить из векторов условий систему из линейно независимых векторов. В таких случаях существенно упрощается определение исходного опорного плана.

Пусть векторы A_1, A_2, \dots, A_m линейно независимы. Разрешим систему уравнений

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

относительно переменных x_1, x_2, \dots, x_m . Получим

$$x_i = x_i^0 + \sum_{j=m+1}^n x_{ij}^0 x_j,$$

где x_i^0 и x_{ij}^0 ($i = 1, \dots, m; j = 1, \dots, n$) — постоянные. Если все $x_i^0 \geq 0$, то, полагая $x_j = 0$ при $j = m+1, \dots, n$, получим опорный план задачи

$$X = (x_1^0, \dots, x_m^0, \underbrace{0, \dots, 0}_{n-m}).$$

Пусть теперь среди x_i^0 имеются отрицательные и пусть x_s^0 — наименьшее из них. Вычтем почленно s -е уравнение из всех уравнений системы. Получим

$$x_i = (x_i^0 - x_s^0) + x_s + \sum_{j=m+1}^n (x_{ij}^0 - x_{sj}^0) x_j,$$

$$x_s = x_s^0 + \sum_{j=m+1}^n x_{sj}^0 x_j.$$

Такое преобразование приводит к базису, составленному из $m-1$ положительных единичных векторов условий задачи и одного искусственного вектора.

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УТОЧНЕНИЯ ОЦЕНОК

Из теории двойственности, изложенной в гл. 3, вытекает ряд методов решения задач линейного программирования. Синтез метода последовательного улучшения плана и основных идей двойственности приводит к методу последовательного уточнения оценок. Впервые этот метод был описан Лемке [26] в 1954 г.

Решение задачи линейного программирования по методу последовательного уточнения оценок сводится к определению оптимального плана сопряженной задачи. Двойственная связь обеих задач позволяет получить при этом и оптимальный план прямой задачи.

В гл. 3 было показано, что компоненты оптимального плана сопряженной задачи являются оценками влияния различных условий задачи линейного программирования на величину максимума линейной формы. Рассматриваемый здесь метод позволяет, исходя из приближенных (предварительных) оценок условий прямой задачи (из начального плана сопряженной задачи), последовательно уточняя их, получить вектор точных оценок условий задачи (оптимальный план сопряженной задачи). С системой предварительных оценок условий, получаемой при каждой итерации метода, может быть связан n -мерный вектор X , который удовлетворяет условиям-равенствам прямой задачи, но может иметь отрицательные компоненты. Системе оценок условий, определяемой в последней итерации, соответствует вектор X^* с неотрицательными составляющими. Вектор X^* является решением прямой задачи. Таким образом, оптимальный опорный план получается в результате последовательного уточнения оценок условий задачи, что и целесообразно отразить в названии метода. Наименование метода, принятое здесь,

больше соответствует его существу, чем название «двойственный симплексный метод», используемое в ряде работ.

Следует заметить, что при переходе от одной итерации к следующей значение линейной формы задачи монотонно убывает. Таким образом, в отличие от метода последовательного улучшения плана, где приближение к максимуму линейной формы происходит снизу, в методе последовательного уточнения оценок приближение к оптимуму осуществляется сверху.

§ 1. ОСНОВЫ МЕТОДА

1.1. Запишем задачу линейного программирования в канонической форме. Требуется вычислить максимум линейной формы

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

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m; \quad (1.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (1.3)$$

Как обычно, будем предполагать ранг матрицы $\|a_{ij}\|_{m,n}$ условий (1.2) равным m . Задача, двойственная по отношению к задаче (1.1)–(1.3) (или сопряженная с ней), состоит в определении минимума линейной формы

$$\tilde{L}(Y) = \sum_{i=1}^m b_i y_i \quad (1.4)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n. \quad (1.5)$$

Более компактно прямая и двойственная задачи записываются в следующем виде:

$$L(X) = (C, X) \rightarrow \max \text{ при условиях } AX^T = B, \quad X \geq 0, \\ \tilde{L}(Y) = (B, Y) \rightarrow \min \text{ при условии } YA \geq C.$$

Напомним применительно к сопряженной задаче (1.4)–(1.5) некоторые понятия, введенные в гл. 3.

План Y сопряженной задачи называется *опорным*, если среди условий (1.5), которые он обращает в равенства, имеется m линейно независимых условий.

Базисом опорного плана Y сопряженной задачи (1.4) — (1.5) назовем произвольную систему из m линейно независимых векторов условий A_{s_i} прямой задачи, для которых

$$(Y, A_{s_\mu}) = c_{s_\mu}, \quad \mu = 1, 2, \dots, m,$$

или, что то же самое,

$$\sum_{i=1}^m a_{is_\mu} y_i = c_{s_\mu}, \quad \mu = 1, 2, \dots, m. \quad (1.6)$$

Обозначим множество индексов (s_1, s_2, \dots, s_m) базиса опорного плана Y через I_Y . Систему векторов A_{s_1}, \dots, A_{s_m} , являющуюся базисом некоторого опорного плана сопряженной задачи, будем для краткости называть *сопряженным базисом*.

Опорный план Y сопряженной задачи называется *невыврожденным*, если для любого вектора A_j , не входящего в его базис,

$$(Y, A_j) > c_j. \quad (1.7)$$

Сопряженная задача (1.4) — (1.5), все опорные планы которой являются невырожденными, называется *невыврожденной* задачей. В ближайших параграфах сопряженная задача будет предполагаться невырожденной.

С каждым опорным планом Y сопряженной задачи (точнее, с его базисом) удобно связывать некоторый n -мерный вектор X , удовлетворяющий условиям (1.2) прямой задачи. Такое соответствие позволит в дальнейшем формулировать в терминах прямой задачи все построения, относящиеся к опорным планам сопряженной задачи.

Разложим вектор ограничений B по сопряженному базису $(A_{s_1}, A_{s_2}, \dots, A_{s_m})$. Обозначим соответствующие коэффициенты разложения через x_{s_i} . Вектор $X = (x_1, \dots, x_n)$, s_i -е компоненты которого ($i = 1, 2, \dots, m$) совпадают с x_{s_i} , а остальные равны нулю, будем называть *псевдопланом* прямой задачи. Компоненты x_{s_i} назовем *базисными* составляющими псевдоплана, а остальные компоненты *внебазисными* переменными. Векторы сопряженного базиса, так

же как и векторы базиса прямой задачи, помимо своих индексов характеризуются номерами позиций, занимаемыми ими в базисе.

Будем обозначать базисную составляющую псевдоплана x_{s_i} через x_{i_0} в соответствии с позицией i , занимаемой вектором A_{s_i} в сопряженном базисе. Ясно, что компоненты псевдоплана удовлетворяют условиям (1.2) прямой задачи

$$\sum_{j=1}^n a_{ij}x_j = \sum_{\mu=1}^m a_{is_{\mu}}x_{s_{\mu}} = b_i, \quad i = 1, 2, \dots, m. \quad (1.8)$$

Следует иметь в виду, что некоторые из $x_{s_{\mu}}$ могут быть отрицательными, так что X не является, вообще говоря, планом прямой задачи. Будем в дальнейшем пользоваться терминами сопряженный базис и базис псевдоплана как синонимами.

Псевдоплан можно определить и независимо от сопряженной задачи. Пусть $S = (A_{s_1}, \dots, A_{s_m})$ — произвольная система линейно независимых векторов условий. Обозначим систему индексов s_1, \dots, s_m через I и разложим векторы условий A_j ($j = 1, 2, \dots, n$) и вектор ограничений $B = A_0$ по векторам системы S :

$$A_j = \sum_{i=1}^m x_{ij}A_{s_i}, \quad (1.9)$$

$$B = A_0 = \sum_{i=1}^m x_{i0}A_{s_i}. \quad (1.10)$$

Введем обозначения

$$\Delta_j = \sum_{i=1}^m c_{s_i}x_{ij} - c_j, \quad j = 1, 2, \dots, n. \quad (1.11)$$

n -мерный вектор X , для которого $x_{s_i} = x_{i_0}$, а $x_j = 0$ при $j \notin I$, является псевдопланом в том и только в том случае, если все $\Delta_j \geq 0$ ($j = 1, 2, \dots, n$).

Действительно, векторы A_{s_1}, \dots, A_{s_m} линейно независимы. Поэтому можно вычислить вектор $Y = (y_1, \dots, y_m)$, составляющие которого удовлетворяют условиям

$$\sum_{\mu=1}^m a_{\mu s_i}y_{\mu} = c_{s_i}.$$

Имеют место следующие соотношения:

$$\begin{aligned}\Delta_j &= \sum_{i=1}^m c_{s_i} x_{ij} - c_j = \sum_{i=1}^m \left(\sum_{\mu=1}^m a_{\mu s_i} y_{\mu} \right) x_{ij} - c_j = \\ &= \sum_{\mu=1}^m y_{\mu} \left(\sum_{i=1}^m a_{\mu s_i} x_{ij} \right) - c_j.\end{aligned}$$

Из формул (1.9) следует, что

$$a_{\mu j} = \sum_{i=1}^m a_{\mu s_i} x_{ij}. \quad (1.12)$$

Поэтому

$$\Delta_j = \sum_{i=1}^m c_{s_i} x_{ij} - c_j = \sum_{\mu=1}^m a_{\mu j} y_{\mu} - c_j. \quad (1.13)$$

Из полученного соотношения вытекает эквивалентность обоих определений псевдоплана.

Следует подчеркнуть, что если план задачи определяется только условиями задачи, то псевдоплан определяется как условиями, так и линейной формой задачи.

Метод последовательного уточнения оценок основывается на применении метода последовательного улучшения плана к сопряженной задаче. Такой подход приводит к новому методу решения прямой задачи.

Рассмотрим некоторый опорный план $Y = (y_1, y_2, \dots, y_m)$ сопряженной задачи. Базис опорного плана Y состоит из векторов A_{s_1}, \dots, A_{s_m} . Ему соответствует псевдоплан X с базисными компонентами x_{s_i} ($i = 1, 2, \dots, m$).

Признак оптимальности. Если среди базисных компонент псевдоплана X нет отрицательных, то псевдоплан X оказывается оптимальным планом прямой задачи, а опорный план Y — решением сопряженной задачи.

Доказательство. Имеет место следующая цепочка равенств:

$$\begin{aligned}\tilde{L}(Y) &= \sum_{i=1}^m b_i y_i = \sum_{(1) i=1}^m \left(\sum_{\mu=1}^m a_{i s_{\mu}} x_{s_{\mu}} \right) y_i = \\ &= \sum_{(2) \mu=1}^m x_{s_{\mu}} \left(\sum_{i=1}^m a_{i s_{\mu}} y_i \right) = \sum_{(3) \mu=1}^m c_{s_{\mu}} x_{s_{\mu}} = \sum_{(4) j=1}^n c_j x_j = L(X).\end{aligned}$$

Равенство (1) следует из соотношения (1.8) для составляющих псевдоплана. Равенство (2) получено

изменением порядка суммирования. Соотношение (3) следует из формулы (1.6). Равенство (4) справедливо потому, что внебазисные компоненты псевдоплана равны нулю.

При неотрицательных x_j псевдоплан X является планом прямой задачи. Согласно лемме 5.2 гл. 3 планы взаимосопряженных задач, связанные соотношением

$$\tilde{L}(Y) = L(X),$$

являются оптимальными планами соответствующих задач. Справедливость признака оптимальности установлена. Заметим, что справедливость сформулированного признака непосредственно следует из второй формы признака оптимальности для метода последовательного улучшения плана, поскольку условия второй формы признака выполняются для псевдоплана X .

В общем случае приведенный признак является только достаточным условием оптимальности плана. Можно доказать, что сформулированный признак оказывается необходимым и достаточным условием оптимальности плана, если исследуемый план сопряженной задачи — невырожденный план.

1.2. Пусть известен некоторый опорный план Y сопряженной задачи с базисом A_{s_1}, \dots, A_{s_m} . Ему соответствует псевдоплан X прямой задачи. Компоненты разложения вектора ограничений $B = A_0$ и векторов условий A_j прямой задачи по векторам сопряженного базиса могут быть вычислены непосредственно из уравнений

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}, \quad j = 0, 1, 2, \dots, n. \quad (1.14)$$

Параметр x_{ij} в формулах (1.14) — это коэффициент разложения вектора условий A_j по векторам сопряженного базиса при векторе A_{s_i} , занимающем i -ю позицию в сопряженном базисе.

В дальнейшем нам понадобится еще один способ вычисления коэффициентов x_{ij} :

$$x_{ij} = \sum_{\mu=1}^m a_{\mu j} e_{i\mu}, \quad i = 1, 2, \dots, m, \quad j = 0, 1, 2, \dots, n, \quad (1.15)$$

где $e_{i\mu}$ — коэффициенты разложения m -мерных единичных векторов e_μ ($\mu = 1, 2, \dots, m$) по векторам сопряженного базиса, т. е.

$$\|e_{ij}\|_m = (A_{s_1}, A_{s_2}, \dots, A_{s_m})^{-1}. \quad (1.16)$$

Формула (1.15) является следствием матричных соотношений

$$A_j = A_Y X_j,$$

или

$$X_j = A_Y^{-1} A_j,$$

где $X_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T$, $A_Y = (A_{s_1}, A_{s_2}, \dots, A_{s_m})$.

В зависимости от знаков базисных составляющих псевдоплана и коэффициентов разложения векторов условий A_j по сопряженному базису следует различать три случая:

1°. Базисные компоненты x_{i0} неотрицательны для всех i ($i = 1, 2, \dots, m$).

2°. Среди x_{i0} имеются отрицательные величины, и по крайней мере для одной из них все $x_{ij} \geq 0$ для $j = 1, 2, \dots, n$.

3°. Псевдоплан содержит отрицательные базисные компоненты x_{i0} , но для каждой из них среди коэффициентов x_{ij} ($j = 1, 2, \dots, n$) имеются отрицательные величины.

В случае 1°, как это следует из признака оптимальности, псевдоплан оказывается оптимальным опорным планом прямой задачи. Ниже мы покажем, что в случае 2° задача линейного программирования неразрешима (условия прямой задачи несовместны), а в случае 3° можно перейти к новому опорному плану сопряженной задачи и, следовательно, к новому псевдоплану с меньшим значением линейной формы.

1.3. Чтобы упростить анализ случаев 2° и 3°, целесообразно проследить за изменением линейной формы (1.4) сопряженной задачи при переходе от опорного плана Y к новому плану

$$Y(\theta) = Y + \theta e^{(i)}. \quad (1.17)$$

Здесь $e^{(i)} = (e_{i1}, \dots, e_{im})$; e_{ij} — определенные выше коэффициенты разложения единичных векторов

$$e_j = \underbrace{(0, \dots, 0, 1, 0, \dots, 0)}_j$$

по векторам базиса плана Y . Чтобы не загромождать запись в обозначении вектора $e^{(i)}$, номер базиса, которому соответствуют составляющие этого вектора, не указывается.

Будем называть переход от плана Y к плану $Y(\theta)$ *элементарным преобразованием* плана Y , связанным с вектором A_{s_i} сопряженного базиса.

Компоненты вектора $Y(\theta) = \{y_1(\theta), \dots, y_m(\theta)\}$ подставим в левую часть уравнений (1.5) сопряженной задачи. Имеем

$$\sum_{\mu=1}^m a_{\mu j} y_{\mu}(\theta) = \sum_{\mu=1}^m a_{\mu j} y_{\mu} + \theta \sum_{\mu=1}^m a_{\mu j} e_{i\mu}.$$

Согласно формулам (1.15)

$$\sum_{\mu=1}^m a_{\mu j} e_{i\mu} = x_{ij}, \quad j = 0, 1, \dots, n.$$

В частности, при $j \in I_Y = (s_1, s_2, \dots, s_m)$

$$\sum_{\mu=1}^m a_{\mu j} e_{i\mu} = \begin{cases} 0 & \text{для } j \in I_Y, j \neq s_i, \\ 1 & \text{для } j = s_i. \end{cases}$$

В соответствии с этими формулами имеем

$$\sum_{\mu=1}^m a_{\mu j} y_{\mu}(\theta) = \begin{cases} \sum_{\mu=1}^m a_{\mu j} y_{\mu}, & j \in I_Y, j \neq s_i, \\ \sum_{\mu=1}^m a_{\mu j} y_{\mu} + \theta, & j = s_i, \\ \sum_{\mu=1}^m a_{\mu j} y_{\mu} + \theta x_{ij}, & j \notin I_Y. \end{cases} \quad (1.18)$$

Выше было введено обозначение (1.11)

$$\Delta_j = \sum_{\mu=1}^m a_{\mu j} y_{\mu} - c_j, \quad j = 1, 2, \dots, n.$$

Поскольку $Y = (y_1, y_2, \dots, y_m)$ — план сопряженной задачи, то

$$\Delta_j \geq 0 \quad \text{для } j = 1, 2, \dots, n. \quad (1.19)$$

Вектор $Y(\theta)$ будет также планом сопряженной задачи в том и только в том случае, если

$$\Delta_j(\theta) = \sum_{\mu=1}^m a_{\mu j} y_{\mu}(\theta) - c_j \geq 0 \quad \text{для } j = 1, 2, \dots, n. \quad (1.20)$$

Определим условия, при которых соотношения (1.20) выполняются. Учитывая формулы (1.11), (1.18) и (1.20), получаем

$$\Delta_j(\theta) = \begin{cases} \Delta_j = 0, & \text{если } j \in I_Y, j \neq s_i, \\ \Delta_j + \theta, & \text{если } j = s_i, \\ \Delta_j + \theta x_{ij}, & \text{если } j \notin I_Y. \end{cases} \quad (1.21)$$

Из (1.21) при $j = s_i$ следует, что $\theta \geq 0$. Если $x_{ij} \geq 0$, то $\Delta_j(\theta) \geq \Delta_j \geq 0$.

Если $x_{ij} < 0$, то для соблюдения условий (1.20) необходимо, чтобы

$$\Delta_j(\theta) = \Delta_j + \theta x_{ij} \geq 0,$$

т. е.

$$\theta \leq -\frac{\Delta_j}{x_{ij}}.$$

Итак, $Y(\theta)$ является планом сопряженной задачи (1.4), (1.5) при всех θ в интервале

$$0 \leq \theta \leq \theta_0, \quad (1.22)$$

где

$$\theta_0 = \min_{x_{ij} < 0} \left(-\frac{\Delta_j}{x_{ij}} \right). \quad (1.23)$$

Если $x_{ij} \geq 0$ для $j = 1, 2, \dots, n$, то θ не ограничено сверху, т. е. $\theta_0 = \infty$. Для невырожденного плана все $\Delta_j > 0$, $j \notin I_Y$. Поэтому $\theta_0 > 0$.

Значение линейной формы $\tilde{L}(Y)$ сопряженной задачи на плане $Y(\theta)$ равно

$$\tilde{L}[Y(\theta)] = \sum_{\mu=1}^m b_{\mu} y_{\mu}(\theta) = \sum_{\mu=1}^m b_{\mu} (y_{\mu} + \theta e_{i_{\mu}}).$$

В соответствии с формулой (1.15) для $j=0$ имеем

$$x_{i0} = \sum_{\mu=1}^m a_{\mu 0} e_{i\mu} - \sum_{\mu=1}^m b_{\mu} e_{i\mu}.$$

Поэтому

$$\tilde{L}[Y(\theta)] = \tilde{L}(Y) + \theta x_{i0}. \quad (1.24)$$

1.4. Приступим к анализу случаев 2° и 3°, выделенных в п. 1.2.

2°. Пусть среди отрицательных компонент x_{i0} псевдоплана X имеется составляющая x_{r0} , для которой $x_{rj} \geq 0$, $j=1, 2, \dots, n$. В этом случае $Y(\theta)$ — результат элементарного преобразования (связанного с вектором A_{sr}) плана Y — является планом сопряженной задачи при любом θ . Следовательно, как видно из соотношения (1.24) (при $i=r$), линейная форма $\tilde{L}(Y)$ не ограничена снизу на множестве планов сопряженной задачи. Согласно лемме 5.3 гл. 3 это возможно лишь в случае несовместности условий прямой задачи.

Заметим, что при исследовании метода последовательного улучшения плана неразрешимость задачи связывалась с неограниченностью линейной формы на множестве планов задачи. Случай несовместности условий задачи не встречался при анализе метода, поскольку мы с самого начала предполагали существование начального плана прямой задачи. При исследовании метода последовательного уточнения оценок мы отправляемся от начального плана сопряженной задачи. Это обеспечивает ограниченность линейной формы прямой задачи, но вовсе не гарантирует совместность ее условий.

3°. Пусть псевдоплан содержит отрицательные базисные компоненты, но для каждой из них имеются отрицательные коэффициенты x_{ij} . В этом случае вектор $Y(\theta)$ будет планом сопряженной задачи при любом $\theta \geq 0$, не превышающем θ_0 из (1.23). Выберем отрицательную базисную компоненту $x_{r0} < 0$ псевдоплана X . По условию среди x_{rj} имеются отрицательные величины. Пусть значение индекса j , на котором достигается θ_0 , равно k :

$$\theta_0 = -\frac{\Delta_k}{x_{rk}}.$$

Очевидно, $\Delta_k(\theta_0) = \Delta_k + \theta_0 x_{rk} = 0$.

В соответствии с теоремой 2.1 гл. 4 система векторов, полученная из сопряженного базиса плана Y заменой вектора A_{s_r} на вектор условий A_k , является линейно независимой системой, так как по условию $x_{rk} \neq 0$ ($x_{rk} < 0$). Следовательно, $Y' = Y(\theta_0)$ является не только планом, но и опорным планом сопряженной задачи. Базис опорного плана Y' получается из системы A_{s_1}, \dots, A_{s_m} заменой A_{s_r} на A_k .

В силу предположения о невырожденности задачи Y' — невырожденный план, т. е. $\Delta_j(\theta_0) > 0$ при $j \notin I_{Y'}$. Отсюда следует, что θ_0 достигается только на одном векторе A_k . Таким образом, в невырожденных случаях вектор, подлежащий включению в базис, определяется однозначно.

Из формулы (1.24) видно, что при переходе от плана Y к $Y' = Y(\theta_0)$ значение линейной формы сопряженной задачи уменьшится на $|\theta_0 x_{r0}| > 0$ ($\theta_0 > 0$, $x_{r0} < 0$).

Заметим, что элементарному преобразованию опорного плана Y в план $Y' = Y(\theta_0)$ соответствует переход от псевдоплана X к псевдоплану X' в прямой задаче.

Значение линейной формы $L(X)$ прямой задачи на псевдоплане равно значению линейной формы $\tilde{L}(Y)$ на соответствующем опорном плане сопряженной задачи. Действительно,

$$\begin{aligned} L(X) &= \sum_{j=1}^n c_j x_j = \sum_{j \notin I_{Y'}} c_j x_j = \sum_{i=1}^m c_{s_i} x_{i0} = \sum_{i=1}^m x_{i0} \sum_{\mu=1}^m a_{\mu s_i} y_{\mu} = \\ &= \sum_{\mu=1}^m y_{\mu} \sum_{i=1}^m a_{\mu s_i} x_{i0} = \sum_{\mu=1}^m y_{\mu} b_{\mu} = \tilde{L}(Y). \end{aligned} \quad (1.25)$$

Таким образом, в случае 3° элементарное преобразование опорного плана Y сопряженной задачи, связанное с вектором A_{s_r} , приводит к уменьшению линейной формы $\tilde{L}(Y)$. В терминах прямой задачи это значит, что в случае 3° можно перейти от псевдоплана X к псевдоплану X' с меньшим значением линейной формы.

Последовательные переходы от одного опорного плана Y сопряженной задачи к другому (или, в терминах прямой задачи, последовательные преобразования одного псевдоплана в другой) производятся до тех пор, пока

будет получено решение задачи или установлена ее неразрешимость.

Каждый переход от одного псевдоплана к следующему составляет итерацию (шаг) метода последовательного уточнения оценок. При использовании этого метода для решения задач с невырожденной сопряженной задачей мы не можем вернуться к уже пройденному раз базису. В противном случае опорные планы сопряженной задачи, полученные при различных итерациях, совпадали бы между собой. Но этого не может быть из-за монотонного убывания линейной формы сопряженной задачи. Таким образом, количество итераций, необходимых для решения невырожденной задачи (или установления ее неразрешимости), заведомо ограничено общим числом базисов сопряженной задачи, не превышающим C_n^m .

В настоящем параграфе метод уточнения оценок излагался независимо от метода улучшения плана. Предлагаем читателю проверить, что метод уточнения оценок представляет собой не что иное, как приложение метода улучшения плана к решению сопряженной задачи.

1.5. Наметим кратко порядок операций, связанных с отдельным шагом метода последовательного уточнения оценок. Каждая итерация содержит два этапа. На первом этапе следует выяснить, не является ли псевдоплан планом прямой задачи, и если нет, то разрешима ли задача. Для этого необходимо вычислить базисные составляющие псевдоплана (разложить вектор ограничений по векторам сопряженного базиса) и установить их знаки. Псевдоплан оказывается планом и, следовательно, решением задачи (случай 1°), если все его компоненты неотрицательны. Если признак оптимальности не удовлетворяется, следует проверить, нет ли оснований утверждать, что условия прямой задачи несовместны. Случай 2° (неограниченность снизу линейной формы сопряженной задачи и, следовательно, противоречивость условий прямой задачи) имеет место, если для некоторой отрицательной базисной переменной псевдоплана $x_{r_0} < 0$ все коэффициенты x_{rj} неотрицательны. И, наконец, случай 3° встречается, если при любом i , для которого $x_{i0} < 0$, среди коэффициентов x_{ij} имеются отрицательные.

Таким образом, первый этап итерации приводит к одной из трех возможностей (случаи 1°, 2°, 3°). Если имеет место случай 1° или 2°, процесс решения заканчивается. В случае 3° переходят ко второму этапу итерации. Вторым этапом является выбор и осуществлении элементарного преобразования, приводящего к новому опорному плану сопряженной задачи и, следовательно, к новому псевдоплану прямой задачи с меньшим значением линейной формы. На втором этапе итерации выбирается позиция r сопряженного базиса с отрицательной компонентой псевдоплана и определяется вектор A_k , который должен заменить в сопряженном базисе исключаемый из него вектор A_{s_r} . Новый сопряженный базис образуется, таким образом, из векторов старого базиса заменой вектора условий A_{s_r} на вектор A_k . На втором этапе вычисляются, кроме того, все параметры, необходимые для получения и исследования нового псевдоплана. Анализ очередного псевдоплана производится на первом этапе следующей итерации.

1.6. Приведем без формальных обоснований геометрическое истолкование псевдоплана, которое может быть использовано для геометрической интерпретации метода последовательного уточнения оценок.

Псевдоплан соответствует точке пересечения n независимых гиперплоскостей, из которых m гиперплоскостей отвечают условиям (1.2) прямой задачи, а $n - m$ соответствуют внебазисным составляющим ($x_j = 0$ для $j \notin I_Y$) псевдоплана. В общем случае точка X , отвечающая псевдоплану X , лежит вне многогранного множества M условий прямой задачи. Чтобы уточнить геометрическое определение псевдоплана, введем многогранный конус K_X , определяемый условиями

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

$$x_j \geq 0, \quad j \notin I_Y = \{s_1, s_2, \dots, s_m\},$$

и имеющий вершину в точке X . Для того чтобы точка X

пересечения n гиперплоскостей

$$\left. \begin{aligned} \sum_{j=1}^n a_{ij}x_j &= b_i, & i &= 1, 2, \dots, m, \\ x_j &= 0, & j &\notin I_V, \end{aligned} \right\} \quad (1.26)$$

представляла собой геометрический аналог псевдоплана, необходимо и достаточно, чтобы соответствующий многогранный конус K_X был расположен в «нижнем» полупространстве, определяемом гиперплоскостью линейной формы, проведенной через точку X . Можно показать, что это условие соответствует неравенствам $\Delta_j \geq 0$ ($j=1, 2, \dots, n$), которые вместе с равенствами (1.26) определяют псевдоплан задачи. Будем в дальнейшем называть точку X — геометрический образ псевдоплана X — псевдовершиной многогранного множества условий прямой задачи (1.1) — (1.3).

Приведенное геометрическое истолкование псевдоплана позволяет сформулировать в терминах первой геометрической интерпретации задачи линейного программирования все элементы отдельной итерации метода последовательного уточнения оценок. Предоставляем это сделать читателю.

В терминах второй геометрической интерпретации псевдоплан X изображается гиперплоскостью, натянутой на m расширенных векторов базиса псевдоплана. Гиперплоскость — образ псевдоплана X — расположена «над» конусом K .

При решении задачи методом последовательного уточнения оценок гиперплоскость, отвечающая очередному псевдоплану, пересекает прямую Q в точке, расположенной ниже точки пересечения прямой Q с образом предыдущего псевдоплана. Через конечное число шагов приходим к гиперплоскости, пересекающей прямую Q в «верхней» точке пересечения прямой Q с конусом K . Таким образом, в методе последовательного уточнения оценок приближение к оптимальному плану осуществляется не изнутри конуса K , как в методе последовательного улучшения плана, а извне.

1.7. Невырожденность двойственной задачи (1.4), (1.5) означает, что любая гиперплоскость линейной формы

(т. е. любая гиперплоскость вида $CX = \text{const}$) пересекает не более одной псевдовершины многогранного множества условий прямой задачи. Предоставляем читателю убедиться, что геометрическое определение невырожденности сопряженной задачи соответствует аналитическому определению, приведенному в п. 1.1.

Исходя из геометрического определения невырожденности задачи (1.4), (1.5), очевидно, что вырожденную двойственную задачу можно свести к невырожденной за счет некоторого достаточно малого изменения коэффициентов линейной формы (1.1). На этом соображении основано построение и обоснование правила однозначного выбора вектора A_k , подлежащего включению в базис. Приведем это правило без доказательств.

Пусть $s_1 < s_2 < \dots < s_m$ — номера векторов условий, образующих базис некоторого псевдоплана задачи (1.1) — (1.3).

Обозначим через E_1 совокупность индексов j , на которых достигается

$$\theta_0 = \min_{x_{rj} < 0} \left(-\frac{\Delta_j}{x_{rj}} \right).$$

Если индекс k (номер вектора, подлежащего включению в базис) еще не определен (E_1 содержит несколько элементов), то приступаем к следующему шагу. Если все индексы $j \in E_1$ превосходят s_m , то $k = \min_{j \in E_1} j$. В противном случае строим множество E_2 , объединяя в него те индексы j , на которых достигается

$$\min_{\substack{j \in E_1 \\ j < s_m}} \frac{x_{mj}}{x_{rj}}.$$

Если множество E_2 содержит только один элемент, то он принимается в качестве искомого индекса k . Если же это условие не выполняется, то приступаем к осуществлению следующего шага, который завершается либо определением k , либо построением множества индексов E_3 , содержащего более одного элемента. Правило проведения данного шага совпадает с уже описанным правилом, с тем лишь отличием, что индекс s_m заменяется на s_{m-1} , а x_{mj} — на $x_{m-1, j}$. Последовательные шаги осуществляются либо

до определения индекса k , либо до построения множества E_{m+1} . В последнем случае индекс k определяется как наименьший из $j \in E_{m+1}$:

$$k = \min_{j \in E_{m+1}} j.$$

Сформулированное правило отыскания индекса k вектора A_k , подлежащего вводу в базис, исключает возможность возвращения к пройденному ранее псевдоплану. Таким образом, использование этого правила дает полную гарантию от заикливания.

§ 2. ПЕРВЫЙ АЛГОРИТМ

2.1. Метод последовательного уточнения оценок может быть, так же как и метод последовательного улучшения плана, реализован в виде двух различных вычислительных схем. Различие между вычислительными схемами (алгоритмами метода) основано на разных способах вычисления параметров Δ_j .

При решении задачи линейного программирования методом последовательного уточнения оценок начальный опорный план сопряженной задачи (точнее, начальный сопряженный базис) предполагается известным.

При использовании первого алгоритма вычисление оптимального плана производится по следующей схеме. Базис начального опорного плана сопряженной задачи принимается в качестве базиса псевдоплана X прямой задачи. Разлагая вектор ограничений по векторам сопряженного базиса, получают базисные компоненты x_{i_0} псевдоплана X . Знаки x_{i_0} позволяют установить, является ли полученный псевдоплан планом задачи. Если случай 1° имеет место, начальный опорный план сопряженной задачи оказывается ее решением, а начальный псевдоплан — оптимальным планом прямой задачи. При наличии отрицательных компонент x_{i_0} следует вычислить коэффициенты x_{ij} разложения векторов условий по сопряженному базису. Если для некоторого $x_{r_0} < 0$ все $x_{rj} \geq 0$, задача неразрешима (случай 2°). Процесс решения задачи заканчивается в этом случае установлением противоречивости ее условий. Если имеет место случай 3°, переходят ко второму этапу итерации. По формулам (1.11) вычисляются параметры Δ_j .

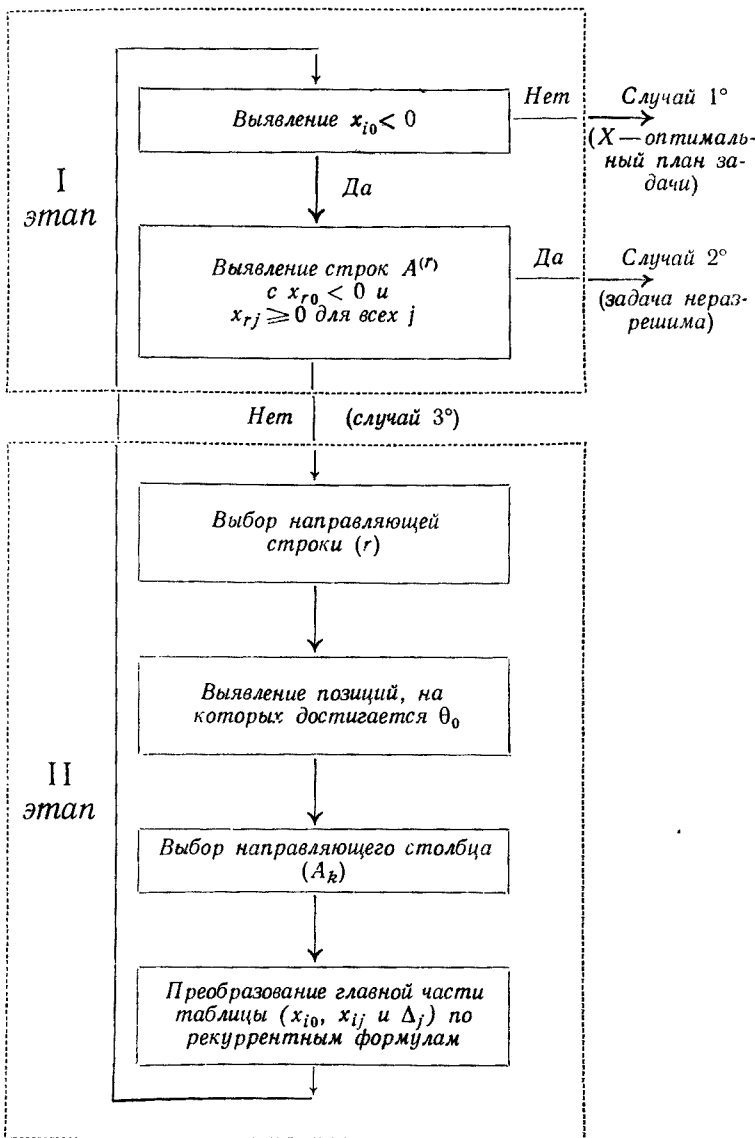


Рис. 5.1.

Затем устанавливаются номера вектора, который должен быть исключен из сопряженного базиса, и вектора условий, который должен занять его место в базисе. С вновь полученным сопряженным базисом на следующей итерации производятся те же операции, что и на предыдущем шаге. Параметры x_{i0} , x_{ij} и Δ_j вычисляются по соответствующим параметрам предыдущего шага по тем же рекуррентным формулам (5.12) гл. 4, что и в первом алгоритме метода последовательного улучшения плана. Вычисления продолжаются до получения оптимального плана или до установления неразрешимости задачи. Процесс решения укладывается в конечное число шагов.

Как видим, вычислительная схема первого алгоритма метода последовательного уточнения оценок весьма схожа с вычислительной схемой первого алгоритма метода последовательного улучшения плана. Близки друг другу и формы записи параметров итераций в таблицах.

Различие между методами по существу заключается в том, что в одном случае производится последовательный переход от одного опорного плана задачи к соседнему, а в другом случае — от одного псевдоплана к очередному псевдоплану. В первом случае мы перемещаемся по базисам прямой задачи, а во втором случае по базисам сопряженной задачи. Формальное различие между вычислительными схемами проявляется только в правилах перехода от одного базиса к следующему и в признаках оптимальности плана и неразрешимости задачи. В методе улучшения плана вначале определяется вектор, подлежащий вводу в базис, а затем — вектор, исключаемый из базиса. В методе уточнения оценок, наоборот, выбор вводимого в базис вектора производится после определения вектора, удаляемого из базиса. Отсюда и некоторое различие в таблицах первых алгоритмов обоих методов.

Блок-схема первого алгоритма метода последовательного уточнения оценок произведена на рис. 5.1.

2.2. Проиллюстрируем процесс решения задач линейного программирования методом последовательного уточнения оценок на числовом примере.

Требуется обратить в максимум линейную форму

$$L(X) = -(3x_1 + x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 5x_7)$$

при условиях

$$\begin{array}{rcl}
 2x_1 + 3x_3 & + 2x_6 & - x_8 = 10, \\
 4x_2 & + 3x_5 + x_8 & - x_9 = 4, \\
 x_2 + 5x_3 & + 4x_5 & + 5x_7 - x_{10} = 2, \\
 3x_1 & + 2x_4 & + 3x_8 - x_{11} = 5, \\
 & 2x_3 + 4x_4 & + 3x_7 - x_{12} = 3, \\
 & x_j \geq 0, & j=1, 2, \dots, 12.
 \end{array}$$

Сопряженная задача формулируется следующим образом. Требуется обратить в минимум линейную форму

$$\tilde{L}(Y) = 10y_1 + 4y_2 + 2y_3 + 5y_4 + 3y_5$$

при условиях

$$\begin{array}{rcl}
 2y_1 & + 3y_4 & \geq -3, \\
 & 4y_2 + y_3 & \geq -1, \\
 3y_1 & + 5y_3 & + 2y_5 \geq -2, \\
 & & 2y_4 + 4y_5 \geq -3, \\
 & 3y_2 + 4y_3 & \geq -1, \\
 2y_1 + y_2 & + 3y_4 & \geq -2, \\
 & 5y_3 & + 3y_5 \geq -5, \\
 & y_j \leq 0, & j=1, 2, 3, 4, 5.
 \end{array}$$

Как видим, $Y = (0, 0, 0, 0, 0)$ является опорным планом сопряженной задачи. Вектор Y обращает в равенства последние $m=5$ условий: $y_j=0$; остальным условиям сопряженной задачи Y удовлетворяет как строгим неравенствам. Базис опорного плана Y сопряженной задачи состоит из векторов условий $A_8 = (-1, 0, 0, 0, 0)$, $A_9 = (0, -1, 0, 0, 0)$, $A_{10} = (0, 0, -1, 0, 0)$, $A_{11} = (0, 0, 0, -1, 0)$ и $A_{12} = (0, 0, 0, 0, -1)$. Все вычисления, связанные с решением примера, приведены в табл. 5.1 (0—3).

Базис исходного псевдоплана состоит из единичных векторов, взятых с обратным знаком. Поэтому базисные компоненты псевдоплана X (таблица 0) совпадают с соответствующими составляющими вектора ограничений, взятыми с обратным знаком, а элементы x_{ij} столбцов A_j равны по величине и противоположны по знаку элементам соответствующих столбцов матрицы условий $\|a_{ij}\|$. В $(m+1)$ -й (шестой) строке таблицы записаны значения $\Delta_j = -c_j$ (базисным компонентам исходного псевдоплана отвечают нулевые коэффициенты линейной формы).

Все базисные компоненты псевдоплана отрицательны и в каждой строке имеются отрицательные коэффициенты x_{ij} ($i=1, 2, 3, 4, 5$). Это значит, что мы имеем дело со случаем 3°. Переходим к новому псевдоплану. Выводу из базиса подлежит вектор $A_{s_1} = A_8$ ($x_8 = x_{1,0} = -10 < x_{10}$, $i=2, 3, 4, 5$). Последняя строка таблицы 0 заполняется значениями $\theta_j = -\frac{\Delta_j}{x_{1j}}$ для $x_{1j} < 0$. Наименьший элемент θ_0 строки θ достигается на векторе A_3 . Вектор условий A_3 подлежит вводу в базис псевдоплана вместо вектора A_8 . Преобразование главной части таблицы 0 в главную часть таблицы 1 производится по рекуррентным формулам (5.12) гл. 4.

Таблица 5.1 (0—3)

		c_j	—												—	
N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	N_T
← 1		A_8	—10	—2		—3			—2		1					
2		A_9	—4		—4			—3	—1			1				
3		A_{10}	—2		—1	—5		—4		—5			1			
4		A_{11}	—5	—3			—2		—3					1		
5		A_{12}	—3			—2	—4			—3					1	
6	—	Δ		3	1	2	3	1	2	5						
7	—	θ	—	1,5	—	0,667	—	—	1	—	—	—	—	—	—	
→ 1	—2	A_3	3,333	0,667		1			0,667		—0,333					
2		A_9	—4		—4			—3	—1			1				
3		A_{10}	14,667	3,333	—1			—4	3,333	—5	—1,667		1			
← 4		A_{11}	—5	—3			—2		—3					1		
5		A_{12}	3,667	1,333			—4		1,333	—3	—0,667				1	
6	—	Δ	—6,667	1,667	1		3	1	0,667	5	0,667					
7	—	θ	—	0,556	—	—	1,5	—	0,222	—	—	—	—	—	—	

0

1

Продолжение табл. 5.1

c_j	—	—	—3	—1	—2	—3	—1	—2	—5							
N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	N_T
1	—2	A_3	2,222			1	—0,444				—0,333				0,222	
2		A_3	—2,333	1	—4		0,667	—3				1			—0,333	
3		A_{10}	9,111		—1		—2,222	—4		—5	—1,667		1		1,111	
4	—2	A_6	1,667	1			—0,667		1						—0,333	
5		A_{12}	1,444				—4,889			—3	—0,667				0,444	1
6	—	Δ	—7,778	1	1		2,556	1		5	0,667				0,222	
7	—	θ	—	—	0,25		—	0,333	—	—	—	—	—	—	0,667	—
1	—2	A_3	2,222			1	—0,444				—0,333				0,222	
2	—1	A_2	0,583	—0,25	1		—0,167	0,75				—0,25			0,083	
3		A_{10}	9,694	—0,25			—2,389	—3,25		—5	—1,667	—0,25	1		1,194	
4	—2	A_6	1,667	1			0,667		1						—0,333	
5		A_{12}	1,444				—4,889			—3	—0,667				0,444	1
6	—	Δ	—8,361	1,25			2,722	0,25		5	0,667				0,139	
7	—	θ	—	—	—	—	—	—	—	—	—	—	—	—	—	—

2

3

2

3

Последующий анализ таблицы 1 и вычисления, связанные с переходом к таблице 2, производятся по аналогичным правилам. Псевдоплан, полученный после третьей итерации, оказывается планом задачи. Таким образом, решением задачи является вектор

$$X''' = (0; 0,583; 2,222; 0; 0; 1,667; 0; 0; 0; 9,694; 0; 1,444).$$

Максимальное значение линейной формы равно

$$L(X''') = -8,361.$$

§ 3. ВТОРОЙ АЛГОРИТМ

Второй алгоритм метода последовательного уточнения оценок отличается от первого алгоритма теми же особенностями, какими второй алгоритм метода последовательного улучшения плана отличается от первого алгоритма, реализующего тот же метод. Решение задачи начинается с заданного начального опорного плана сопряженной задачи. Вычисления записываются так же, как и во втором алгоритме метода улучшения плана: в серии основных и одной вспомогательной таблице.

Блок-схема второго алгоритма метода последовательного уточнения оценок приведена на рис. 5.2.

Сравнение первого и второго алгоритмов метода последовательного уточнения оценок приводит к тем же выводам, что и сравнение обоих алгоритмов метода улучшения плана.

Проиллюстрируем решение задачи линейного программирования вторым алгоритмом метода последовательного уточнения оценок на примере, рассмотренном в п. 2.2 при изложении первого алгоритма.

Весь ход вычислений зафиксирован в основной табл. 5.2 (3-й) и вспомогательной табл. 5.3.

Примем в качестве исходного сопряженного базиса ту же систему векторов, что и при решении задачи первым алгоритмом. Базисные переменные не входят в линейную форму задачи. Поэтому элементы строки Δ вспомогательной таблицы совпадают с соответствующими коэффициентами линейной формы, взятыми с обратным знаком. Из базиса выводится вектор $A_{s_1} = A_8$, отвечающий наибольшей по абсолютной величине отрицательной компоненте псевдоплана.

Строка X_1 , элементы которой вычисляются по формуле

$$x_{rj} = \sum_{i=1}^m e_{ri} a_{ij}, \quad j = 1, 2, \dots, n,$$

содержит отрицательные величины. Следовательно, мы имеем дело со случаем 3°.

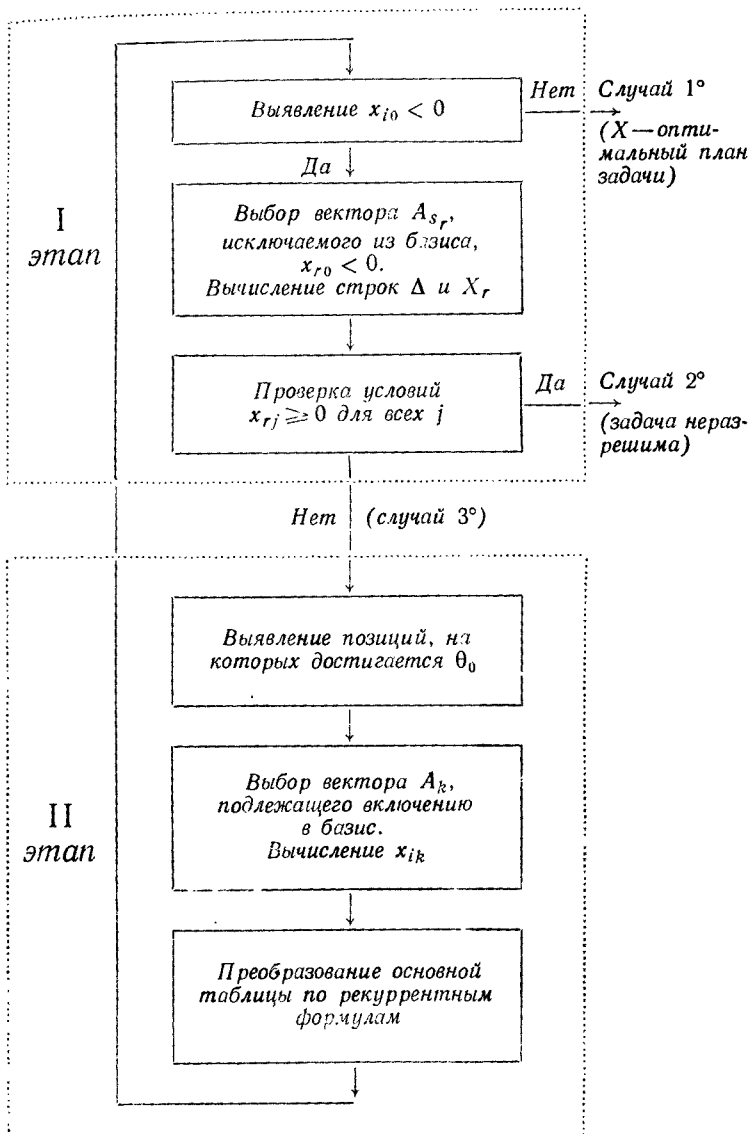


Рис. 5.2.

Таблица 5.2 (0—3)

	N	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	$A_k^{(l)}$	N_T
←	1		A_8	-10	-1					-3	0
	2		A_9	-4		-1					
	3		A_{10}	-2			-1			-5	
	4		A_{11}	-5				-1			
	5		A_{12}	-3					-1	-2	
	6	—	L							2	
→	1	-2	A_3	3,333	0,333					0,667	1
	2		A_9	-4		-1				-1	
	3		A_{10}	14,667	1,667		-1			3,333	
←	4		A_{11}	-5				-1		-3	
	5		A_{12}	3,667	0,667				-1	1,333	
	6	—	L	-6,667	-0,667					0,667	
	1	-2	A_3	2,222	0,333			-0,222			2
←	2		A_9	-2,333		-1		0,333		-4	
	3		A_{10}	9,111	1,667		-1	-1,111		-1	
→	4	-2	A_6	1,667				0,333			
	5		A_{12}	1,444	0,667			-0,444	-1		
	6	—	L	-7,778	-0,667			-0,222		1	
	1	-2	A_3	2,222	0,333			-0,222			3
→	2	-1	A_2	0,583		0,25		-0,0833			
	3		A_{10}	9,694	1,667	0,25	-1	-1,194			
	4	-2	A_6	1,667				0,333			
	5		A_{12}	1,444	0,667			-0,444	-1		
	6	—	L	-8,361	-0,667	-0,25		-0,139			

Таблица 5.3

N	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
1	10	2		3			2		-1				
2	4		4			3	1			-1			
3	2		1	5		4		5			-1		
4	5	3			2		3					-1	
5	3			2	4			3					-1
6	c_j	-3	-1	-2	-3	-1	-2	-5					
0	Δ	3	1	2	3	1	2	5					
	X_1	-2		-3			-2		1				
	θ	1,5	-	0,667	-	-	1	-	-	-	-	-	-
1	Δ'	1,667	1		3	1	0,667	5	0,667				
	X'_4	-3			-2		-3					1	
	θ'	0,556	-	-	1,5	-	0,222	-	-	-	-	-	-
	Δ''	1	1		2,556	1		5	0,667			0,222	
2	X''_2	1	-4		0,667	-3				1		-0,333	
	θ''	-	0,25	-	-	0,333	-	-	-	-	-	0,667	-

В строке θ заполняются лишь позиции 1, 3 и 6, которые соответствуют отрицательным элементам строки X_1 . Наименьший элемент строки θ достигается на векторе A_3 . Последний подлежит включению в первую позицию базиса вместо вектора A_8 . Коэффициенты разложения A_3 по исходному базису вычисляются по формулам

$$x_{ik} = \sum_{s=1}^m e_{is} a_{sk}, \quad i=1, 2, \dots, m,$$

и записываются в столбец $A_k^{(l)}$ основной таблицы 0. В правый нижний угол таблицы записывается значение $\Delta_k = \Delta_3 = 2$. После этого главная часть таблицы 0 преобразуется в главную часть основной таблицы 1 по рекуррентным формулам

$$e'_{1j} = \frac{1}{x_{13}} e_{1j}, \quad e'_{ij} = e_{ij} - e'_{1j} x_{i3}, \\ i=1, 2, 3, 4, 5, \quad j=0, 1, 2, 3, 4, 5.$$

Например,

$$e'_{11} = \frac{1}{-3} \cdot (-1) = 0,333, \quad e'_{51} = 0 - 0,333 \cdot (-2) = 0,667.$$

Приступаем к первому этапу итерации 1. Столбец e_0 снова содержит отрицательные элементы. Исключению из базиса подлежит $A_{s_4} = A_{11}$. Строка Δ' вспомогательной таблицы вычисляется по рекуррентной формуле

$$\Delta_j^{(j+1)} = \Delta_j^{(l)} + \theta_0^{(l)} x_{rj}^{(l)}, \quad j=0, 1, 2, \dots, n.$$

Имеем, например,

$$\Delta'_8 = \Delta_8 + \theta_0 x_{18} = 2 + 0,667 \cdot (-2) = 0,667.$$

Строка X'_4 вспомогательной таблицы содержит отрицательные элементы. Следовательно, нет оснований для суждений о неразрешимости задачи.

Переходим ко второму этапу итерации 1. Заполняем строку θ' . Наименьший элемент этой строки $\theta'_0 = 0,222$ достигается на векторе A_6 . Вектор условий A_6 вводится в четвертую позицию очередного базиса вместо вектора $A_{s_4} = A_{11}$. В последний столбец $A_k^{(l)}$ основной таблицы 1 вносятся вычисляемые по формуле

$$x_{ik}^{(l)} = \sum_{s=1}^m e_{is}^{(l)} a_{sk}$$

коэффициенты x'_{i8} разложения вектора A_6 по текущему базису. В $(m+1)$ -й (шестой) позиции столбца $A_k^{(l)}$ указывается значение $\Delta'_8 = 0,667$. Основная таблица 1 целиком заполнена и подготовлена к преобразованию ее главной части в главную часть основной таблицы 2. Продолжение той же процедуры приводит после третьей итерации к псевдоплану X'' с положительными составляющими.

План X^* является решением задачи. Базисные составляющие оптимального опорного плана задачи записаны в столбце e_0 основной таблицы 3. Максимальное значение линейной формы прямой задачи и составляющие решения сопряженной задачи содержатся в последней строке основной таблицы 3.

§ 4. СПОСОБЫ ОПРЕДЕЛЕНИЯ ИСХОДНОГО ОПОРНОГО ПЛАНА СОПРЯЖЕННОЙ ЗАДАЧИ

4.1. Решение задачи линейного программирования методом последовательного уточнения оценок начинается с известного сопряженного базиса.

В ряде задач конкретный вид ограничений или физический смысл условий задачи позволяет без труда построить исходный опорный план сопряженной задачи, а следовательно, и сопряженный базис.

Рассмотрим класс задачи линейного программирования, в которых начальный сопряженный базис может быть определен без каких бы то ни было вычислений.

Пусть задача линейного программирования задана в следующем виде.

Требуется определить максимум линейной формы

$$L(X) = \sum_{j=1}^n c_j x_j \quad (4.1)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, 2, \dots, m, \quad (4.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (4.3)$$

Пусть при этом все коэффициенты c_j ($j = 1, 2, \dots, n$) линейной формы (4.1) неположительны.

Задача сводится к канонической форме, если заменить условия (4.2) — (4.3) следующими:

$$\sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i, \quad i = 1, 2, \dots, m, \quad (4.4)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n + m. \quad (4.5)$$

Задача, сопряженная с задачей (4.1), (4.4), (4.5), формулируется следующим образом. Требуется вычислить

минимум линейной формы

$$\tilde{L}(Y) = \sum_{i=1}^m b_i y_i \quad (4.6)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j=1, 2, \dots, n, \quad (4.7)$$

$$-y_i \geq 0, \quad i=1, 2, \dots, m. \quad (4.8)$$

Ясно, что вектор

$$Y = (0, \dots, 0)$$

является опорным планом сопряженной задачи (4.6)—(4.8). Если все $c_j < 0$, то опорный план Y является невырожденным планом. Сопряженный базис составляется в данном случае из векторов

$$A_{n+i} = (\underbrace{0, \dots, 0}_{t}, \underbrace{-1, 0, \dots, 0}_{m}).$$

Таким образом, векторы A_{n+i} ($i=1, 2, \dots, m$) образуют базис исходного псевдоплана. Базисные составляющие начального псевдоплана равны соответствующим компонентам вектора ограничений, взятым с обратным знаком. Точно так же коэффициенты x_{ij} разложения векторов условий A_j по начальному базису псевдоплана равны соответствующим составляющим этих векторов, взятым с обратным знаком.

4.2. Рассмотрим теперь класс задач, в которых исходный опорный план не может быть так просто указан, но определение неопорного плана не представляет труда.

Пусть в задаче (1.1)—(1.3) одна и та же компонента a_{tj} всех векторов условий A_j положительна. Рассмотрим вектор

$$Y = (\underbrace{0, \dots, 0}_{t}, y_t, \underbrace{0, \dots, 0}_{m}), \text{ где } y_t \geq \max_{1 \leq j \leq n} \frac{c_j}{a_{tj}}.$$

Вектор Y удовлетворяет всем условиям задачи (1.4), (1.5) и, следовательно, является планом сопряженной

задачи. К сожалению, вектор Y обычно не является опорным планом. В связи с этим представляет интерес изучение метода, позволяющего по произвольному неопорному плану задачи построить ее опорный план.

Поясним в геометрических терминах идею метода. Система неравенств — условия сопряженной задачи — высекает в m -мерном пространстве переменных y_1, y_2, \dots, y_m многогранное множество S — множество планов сопряженной задачи. Каждый опорный план соответствует вершине множества S .

Пусть задан некоторый план Y' задачи, не совпадающий ни с одной из вершин S . Точка Y' расположена внутри некоторой грани S_1 многогранного множества S (S_1 может совпадать с S). Линейная форма сопряженной задачи порождает на S_1 некоторую линейную функцию. Направление градиента этой функции совпадает с проекцией вектора ограничений B на грань S_1 . Будем передвигать точку Y' в направлении наискорейшего убывания линейной функции до пересечения с границей S_1 . Вновь полученный таким образом план Y'' лежит внутри грани S_2 , размерность которой строго меньше размерности S_1 . Проектируя направляющий вектор гиперплоскости линейной формы \tilde{L} на грань S_2 , получим направление наискорейшего убывания линейной функции, которую \tilde{L} определяет на S_2 .

Продолжая тем же путем процесс дальше и понижая от шага к шагу размерность граней, мы в конце концов попадаем в грань нулевой размерности — в некоторую вершину многогранного множества S . Эта вершина соответствует опорному плану сопряженной задачи. При этом, как правило, полученный опорный план оказывается достаточно хорошим приближением к решению задачи. Это и ясно: движение от Y' к Y'' , от Y'' к Y''' и т. д. шло в направлении наискорейшего убывания линейной формы \tilde{L} на соответствующих гранях множества S .

В процессе движения по указанному методу можно попасть на грань S_p , параллельную гиперплоскости линейной формы. На этой грани значение линейной формы постоянно. Дальнейшее перемещение возможно, вообще говоря, по любому направлению, обеспечивающему выход

на границу этой грани. Если грань S_p представляет собой многогранник, то прямолинейное движение в любом направлении приведет к образу плана, расположенному на грани меньшей размерности. Выбор направления перемещения точки $Y^{(p)}$ несколько ограничивается, если грань S_p — неограниченное многогранное множество. Но и в этом случае всегда можно обеспечить выход на границу рассматриваемой грани.

В [51], § 5 гл. 9 приведенные геометрические рассуждения облечены в аналитическую форму.

4.3. При определении исходного опорного плана методом, указанным в п. 4.2, предполагалось наличие некоторого неопорного плана задачи. Как уже отмечалось, для широкого класса задач вычисление неопорного плана не представляет труда. Тем не менее можно указать и такие задачи, в которых вычисления, связанные с определением исходного плана, не менее трудоемки, чем вычисление опорного плана по неопорному. В таких случаях целесообразно воспользоваться изложенным ниже приемом.

Дополним условия (1.2) задачи линейного программирования ограничением

$$x_1 + x_2 + \dots + x_n \leq M$$

или

$$x_0 + x_1 + \dots + x_n = M, \quad x_0 \geq 0.$$

Будем называть вновь полученную задачу (с переменными x_0, x_1, \dots, x_n) *расширенной задачей*. Сопряженная с ней задача состоит в отыскании минимума линейной формы

$$My_0 + \sum_{i=1}^m b_i y_i \quad (4.9)$$

при условиях

$$y_0 + \sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n, \quad y_0 \geq 0.$$

Назовем эту задачу *расширенной сопряженной задачей*. Очевидно, что в качестве плана расширенной сопряженной

задачи можно взять

$$Y = (y_0, 0, \dots, 0),$$

где

$$y_0 = \max_j \{c_j, 0\}.$$

Отправляясь от плана Y , можно с помощью метода, рассмотренного в п. 4.2, получить опорный план последней задачи. Примем найденный сопряженный базис в качестве базиса исходного псевдоплана и решим расширенную задачу методом последовательного уточнения оценок. В процессе решения величина M предполагается достаточно большой (больше любого числа, с которым ее приходится по ходу решения сравнивать).

Пусть $\bar{X}^* = (x_0^*, x_1^*, \dots, x_n^*)$ и $\bar{Y}^* = (y_0^*, y_1^*, \dots, y_m^*)$ — оптимальные планы прямой и сопряженной расширенных задач. Возможны два случая: $y_0^* = 0$ и $y_0^* > 0$. В первом случае $Y^* = (y_1^*, \dots, y_m^*)$ является, очевидно, решением сопряженной задачи (1.4), (1.5). Поэтому $X^* = (x_1^*, \dots, x_n^*)$ — оптимальный план задачи (1.1) — (1.3)

$$\left(\sum_{j=1}^n c_j x_j^* + 0 \cdot x_0^* = \sum_{i=1}^m b_i y_i^* + M \cdot 0 \right).$$

Во втором случае ($y_0^* > 0$) множество планов сопряженной задачи пусто. Действительно, предположим противное и пусть $Y = (y_1, \dots, y_m)$ — план задачи (1.4), (1.5). Тогда $\bar{Y} = (0, y_1, \dots, y_m)$ — план расширенной задачи. При достаточно большом M значение линейной формы (4.9) в точке $\bar{Y}^* = (y_0^*, y_1^*, \dots, y_m^*)$ превосходит ее значение в точке $\bar{Y} = (0, y_1, \dots, y_m)$. А это противоречит оптимальности плана Y^* . Итак, при $y_0^* > 0$ задача (1.4), (1.5), а следовательно, и сопряженная с ней задача (1.1) — (1.3) неразрешимы.

Если процесс решения расширенной задачи завершился случаем 2°, означающим несовместность ее условий, то это указывает на противоречивость условий исходной задачи. Действительно, если бы исходная задача имела план (x_1, x_2, \dots, x_n) , то при

$$M \geq \sum_{j=1}^n x_j$$

вектор $(0, x_1, x_2, \dots, x_n)$ оказался бы планом расширенной задачи.

Таким образом, рассмотренный метод позволяет найти оптимальный план разрешимой задачи без предварительного вычисления плана сопряженной задачи.

§ 5. МЕТОД УЛУЧШЕНИЯ ПЛАНА И МЕТОД УТОЧНЕНИЯ ОЦЕНОК

5.1. Вычислительные схемы обоих методов близки по форме, и число операций, необходимых для проведения одной итерации в том и другом методе, — величины одного и того же порядка. В общем случае нет обоснованных соображений, позволяющих сравнить число итераций, необходимых для решения задачи каждым из описанных методов.

Особенности методов, изложенные в гл. 4 и 5, позволяют в каждом конкретном случае, исходя из специфики задачи (или класса задач), отдать предпочтение тому или другому методу. При выборе метода важную роль играет трудоемкость определения исходного опорного плана. Если, например, начальный опорный план сопряженной задачи очевиден, а исходный опорный план прямой задачи подлежит определению по общим правилам, то естественно решать задачу по методу уточнения оценок. Ниже будет указана дополнительная особенность метода уточнения оценок, полезная для решения широкого класса задач.

Во многих случаях появляется необходимость решать ряд задач линейного программирования, различающихся одним или несколькими условиями. Корректная постановка сложных практических проблем, исследуемых методами линейного программирования, требует обычно решения целой серии задач, различающихся небольшим числом ограничений. Дело в том, что при обсуждении новых вопросов, в постановке и решении которых нет достаточного опыта, не всегда удается сразу установить все ограничения, которым должно удовлетворять решение задачи. Обычно анализ решения задачи в предварительной постановке приводит к необходимости учета дополнительных условий. Этот процесс нередко повторяется не один раз. Такие же трудности возникают и при решении классических задач линейного программирования.

Например, при составлении экономного плана перевозок (при решении так называемой транспортной задачи) обычно вначале не учитывают ограниченных пропускных способностей коммуникаций. Кроме того, со временем пропускные способности путей сообщения меняются. Естественно потребовать, чтобы корректировка плана была связана с меньшей трудоемкостью, чем составление плана перевозок.

Необходимость в решении серии задач линейного программирования, отличающихся только одним условием, возникает при использовании методов целочисленного программирования.

Естественно, что во всех указанных случаях нецелесообразно при добавлении отдельных условий начинать решать задачу заново. Возникает необходимость в разумном учете полезной информации, заключенной в оптимальном плане первоначальной задачи. Метод последовательного уточнения оценок может быть с успехом использован для этой цели.

Пусть в результате решения исходной задачи (будем ее по-прежнему называть задачей (A)) получен опорный оптимальный план $X^* = (x_1^*, \dots, x_n^*)$ с базисом A_{s_1}, \dots, A_{s_m} . Поставим новую задачу, отличающуюся от задачи (A) дополнительным условием

$$\sum_{j=1}^n a_{m+1, j} x_j \leq b_{m+1},$$

или, что то же самое,

$$\sum_{j=1}^n a_{m+1, j} x_j + x_{n+1} = b_{m+1}, \quad x_{n+1} \geq 0.$$

Назовем эту задачу задачей (C). Сформулируем задачу (\tilde{C}), сопряженную с (C).

Требуется обратить в минимум линейную форму

$$\tilde{L}(Y) = \sum_{i=1}^{m+1} b_i y_i$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i + a_{m+1, j} y_{m+1} \geq c_j, \quad j = 1, 2, \dots, n,$$

$$y_{m+1} \geq 0.$$

Если

$$\sum_{j=1}^n a_{m+1,j} x_j^* \leq b_{m+1}, \quad (5.1)$$

то вектор X^* является, очевидно, решением задачи (С). Пусть теперь условие (5.1) не выполняется, т. е.

$$\sum_{j=1}^n a_{m+1,j} x_j^* > b_{m+1}.$$

Будем решать задачу (С) методом уточнения оценок \bar{y} .

Допустим, что задача (А) решалась вторым алгоритмом метода улучшения плана или метода уточнения оценок. И в том и в другом случае в последней основной таблице содержатся:

а) оптимальный план $X^* = (x_1^*, \dots, x_n^*)$ исходной задачи;

б) оптимальный план $Y^* = (y_1^*, \dots, y_m^*)$ сопряженной задачи;

в) матрица $\|e_{ij}^*\|_m = (A_{s_1}, A_{s_2}, \dots, A_{s_m})^{-1}$, обратная к матрице базиса оптимального плана.

Из условий задачи (\bar{C}) следует, что вектор $\bar{Y}^* = (y_1^*, \dots, y_m^*, 0)$ является ее опорным планом с базисом $\bar{A}_{s_1}, \dots, \bar{A}_{s_m}, \bar{A}_{n+1}$. Здесь по определению

$$\begin{aligned} \bar{A}_j &= (a_{1j}, \dots, a_{mj}, a_{m+1,j}), \quad j = 1, 2, \dots, n, \\ \bar{A}_{n+1} &= (0, 0, \dots, 0, 1). \end{aligned}$$

Следовательно, при решении задачи (\bar{C}) методом уточнения оценок вектор \bar{Y}^* может быть принят в качестве ее исходного опорного плана. Соответствующая ему основная таблица без труда образуется из последней основной таблицы задачи (А). Предоставляем читателю проверить справедливость следующих соотношений:

$$\bar{X}^* = (x_1^*, \dots, x_n^*, x_{n+1}^*), \quad (5.2)$$

где

$$\begin{aligned}x_{n+1}^* &= - \sum_{j=1}^n a_{m+1, j} x_j^* + b_{m+1}, \quad \bar{Y}^* = (y_1^*, \dots, y_m^*, 0), \\ \bar{e}_{ij}^* &= e_{ij}^*, \quad i, j = 1, 2, \dots, m, \\ \bar{e}_{i, m+1}^* &= 0, \quad i = 1, 2, \dots, m, \\ \bar{e}_{m+1, j}^* &= - \sum_{\lambda=1}^m e_{\lambda j}^* a_{m+1, s_\lambda}, \quad j = 1, 2, \dots, m, \\ \bar{e}_{m+1, m+1}^* &= 1.\end{aligned}$$

Черта, поставленная над обозначениями параметров, указывает здесь на принадлежность их к задаче (С). Обычно для получения оптимального плана усложненной задачи требуется небольшое число итераций метода уточнения оценок. Объясняется это тем, что план \bar{Y}^* почти всегда оказывается близким к решению задачи (\tilde{C}) .

5.2. При решении задач линейного программирования часто удается существенно сократить число итераций, если можно удовлетвориться планом, на котором линейная форма отличается от своего оптимального значения не более чем на заданную величину. Поэтому важно уметь оценить проигрыш в величине линейной формы задачи, возникающий за счет замены оптимального плана некоторым промежуточным планом задачи. Такую оценку можно получить, решая одновременно прямую и сопряженную задачи, или, что то же самое, решая задачу одновременно методом улучшения плана и методом уточнения оценок.

Рассмотрим пару взаимно сопряженных задач линейного программирования. Пусть $X = (x_1, \dots, x_n)$ и $Y = (y_1, \dots, y_m)$ — некоторые опорные планы этих задач. Обозначим через V максимальное значение линейной формы прямой задачи. Как известно,

$$V \leq \tilde{L}(Y) = \sum_{i=1}^m b_i y_i.$$

Следовательно,

$$V - L(X) \leq \tilde{L}(Y) - L(X). \quad (5.3)$$

Таким образом, имея некоторый план сопряженной задачи, можно оценить отклонение значения линейной формы прямой задачи на произвольном плане от ее оптимального значения. Согласно первой теореме двойственности оптимальные значения линейных форм прямой и сопряженной задач совпадают. Поэтому оценка (5.3) будет тем точнее, чем ближе Y к решению сопряженной задачи.

Допустим теперь, что к исходной задаче применяются одновременно оба метода: метод улучшения плана, начиная с плана X , и метод уточнения оценок, начиная с плана Y . В результате образуются две последовательности планов:

$$X, X_1, X_2, \dots, X_k; \quad Y, Y_1, Y_2, \dots, Y_k.$$

Согласно (5.3) имеем

$$\delta_k = V - \sum_{j=1}^n c_j x_j^{(k)} \leq \sum_{i=1}^m b_i y_i^{(k)} - \sum_{j=1}^n c_j x_j^{(k)} = \tilde{\delta}_k.$$

Если величина $\tilde{\delta}_k$ укладывается в заранее заданные пределы, процесс решения прекращается, и в качестве оптимального плана принимается план X_k . В противном случае следует последовательно переходить к новым парам планов до тех пор, пока число $\tilde{\delta}_k$ не станет меньше заданной величины.

Указанный комбинированный метод особенно эффективен в применении к задачам линейного программирования с большим числом переменных и ограничений.

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО СОКРАЩЕНИЯ НЕВЯЗОК

Метод последовательного улучшения плана оперирует лишь с исходной задачей. Метод последовательного уточнения оценок имеет дело только с сопряженной задачей. Естественно рассмотреть метод решения задач линейного программирования, использующий обе задачи двойственной пары. Таким методом является метод последовательного сокращения невязок. В этом методе оптимальный план достигается при движении по векторам с неотрицательными компонентами. Правила перехода от одного вектора X к другому обеспечивают сокращение от шага к шагу разностей между правыми и левыми частями условий-равенств исходной задачи, записанной в канонической форме. Эти разности принято называть *невязками*. Доказывается, что через конечное число шагов невязки будут сведены к нулю или будет установлена неразрешимость задачи. Определяющая особенность метода — последовательное сокращение невязок — отражена в его названии.

Идея метода была впервые высказана в 1939 г. Л. В. Канторовичем, применившим ее к решению ряда частных задач линейного программирования. В работах Л. В. Канторовича не рассматривается сопряженная задача, однако введенные им разрешающие множители (см. § 7 гл. 3) определяют ее оптимальный план.

Независимое описание метода сокращения невязок было дано в 1956 г. тремя американскими авторами [16]. История появления этого метода в американской литературе такова. Вначале на основе некоторых соображений, содержащихся в одной старой работе венгерского математика Эгервари [46], метод сокращения невязок был развит для транспортной задачи [25, 40 и 29]. В честь

Эгервари метод был назван венгерским. В работе [16] венгерский метод решения транспортной задачи перенесен на случай общей задачи линейного программирования.

§ 1. ОБЩАЯ СХЕМА МЕТОДА

1.1. Запишем задачу линейного программирования в канонической форме. Требуется вычислить максимум линейной формы

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

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (1.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (1.3)$$

Прямую задачу (1.1) — (1.3) будем называть задачей (A). Задача (\tilde{A}), сопряженная с задачей (A), состоит в определении минимума линейной формы

$$\tilde{L}(Y) = \sum_{i=1}^m b_i y_i \quad (1.4)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n. \quad (1.5)$$

Будем предполагать все компоненты вектора ограничений $B = (b_1, b_2, \dots, b_m)^T$ неотрицательными. Ясно, что каждая задача линейного программирования может быть приведена к эквивалентной задаче, удовлетворяющей этому условию.

Изложение метода может быть упрощено, если ввести в рассмотрение так называемые *расширенную* и *вспомогательные* задачи. Будем для краткости называть расширенную задачу задачей (B). Задача (B) состоит в определении минимума линейной формы

$$\sum_{i=1}^m \varepsilon_i \quad (1.6)$$

при соблюдении ограничений

$$\sum_{j=1}^n a_{ij}x_j + \varepsilon_i = b_i, \quad i = 1, 2, \dots, m, \quad (1.7)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n, \quad (1.8)$$

$$\varepsilon_i \geq 0, \quad i = 1, 2, \dots, m. \quad (1.9)$$

Пусть $Y = (y_1, y_2, \dots, y_m)$ — некоторый план сопряженной задачи (\bar{A}) (не обязательно опорный). Выделим векторы условий A_j , $j = 1, 2, \dots, n$, для которых

$$\sum_{i=1}^m a_{ij}y_i = c_j. \quad (1.10)$$

Обозначим множество индексов таких векторов через E_Y . С каждым планом Y сопряженной задачи (\bar{A}) может быть связана вспомогательная задача (C_Y) , образующаяся из расширенной задачи (B) введением дополнительных ограничений

$$x_j = 0 \quad \text{при} \quad j \notin E_Y. \quad (1.11)$$

Таким образом, во вспомогательной задаче (C_Y) требуется обратить в минимум линейную форму

$$\sum_{i=1}^m \varepsilon_i \quad (1.12)$$

при условиях

$$\sum_{j \in E_Y} a_{ij}x_j + \varepsilon_i = b_i, \quad (1.13)$$

$$x_j \geq 0 \quad \text{при} \quad j \in E_Y, \quad (1.14)$$

$$\varepsilon_i \geq 0, \quad i = 1, 2, \dots, m. \quad (1.15)$$

В процессе решения задачи (A) методом последовательного сокращения невязок понадобится также задача, двойственная по отношению к вспомогательной задаче (C_Y) . Назовем сопряженную вспомогательную задачу задачей (\tilde{C}_Y) . Согласно общим правилам построения двойственных задач задача (\tilde{C}_Y) формулируется следующим образом.

Требуется обратить в максимум линейную форму

$$\sum_{i=1}^m b_i \mu_i, \quad (1.16)$$

переменные которой удовлетворяют условиям

$$\sum_{i=1}^m a_{ij} \mu_i \leq 0 \quad \text{при } j \in E_Y, \quad (1.17)$$

$$\mu_i \leq 1, \quad i = 1, 2, \dots, m. \quad (1.18)$$

Векторами условий вспомогательной задачи (C_Y) являются векторы $A_{j_1}, A_{j_2}, \dots, A_{j_s}, e_1, e_2, \dots, e_m$, где индексы j_1, \dots, j_s составляют множество E_Y , а векторы e_1, e_2, \dots, e_m образуют полный набор m -мерных единичных векторов.

Пусть компоненты произвольного плана вспомогательной задачи (C_Y) равны соответственно ξ_j ($j \in E_Y$), ε_i ($i = 1, 2, \dots, m$). Поставим в соответствие каждому такому плану n -мерный вектор $X = (x_1, x_2, \dots, x_n)$, где

$$\left. \begin{aligned} x_j &= \xi_j & \text{при } j \in E_Y, \\ x_j &= 0 & \text{при } j \notin E_Y. \end{aligned} \right\} \quad (1.19)$$

Вектор X с неотрицательными компонентами, определенный формулами (1.19), вообще говоря, не является планом исходной задачи (A). В соответствии с соотношениями (1.13), (1.15), (1.19)

$$b_i - \sum_{j=1}^n a_{ij} x_j = \varepsilon_i \geq 0. \quad (1.20)$$

Компоненты ε_i плана вспомогательной задачи определяют, таким образом, невязки, возникающие в i -м условии системы (1.2) при подстановке в нее вектора $X = (x_1, \dots, x_n)$.

Оптимальный опорный план вспомогательной задачи содержит не более m положительных компонент (ξ_j, ε_i). Число положительных компонент соответствующего вектора X тем более не превышает m .

Будем называть вектор X , отвечающий оптимальному опорному плану некоторой вспомогательной задачи (C_Y), *квазипланом* исходной задачи (A). Векторы базиса решения вспомогательной задачи образуют также *базис квазиплана* задачи (A). Базис B_X квазиплана состоит из некоторого числа векторов условий задачи (A) и нескольких единичных векторов. Если в базисе квазиплана нет единичных векторов, то квазиплан оказывается планом задачи (A) и, как мы увидим далее, ее решением.

Вектор $E = (\varepsilon_1, \dots, \varepsilon_m)$, где $\varepsilon_1, \dots, \varepsilon_m$ — последние m составляющих решения вспомогательной задачи, будем называть *вектором невязок*, соответствующим квазиплану X , а сумму $\varepsilon_0 = \sum_{i=1}^m \varepsilon_i$ компонент вектора E — *невязкой квази-плана*.

Признак оптимальности. Квазиплан X задачи (A) является ее оптимальным планом, если все компоненты соответствующего вектора невязок равны нулю, или, что то же самое, квазиплан задачи с нулевой невязкой является ее решением.

Доказательство. По построению компоненты x_j квазиплана неотрицательны. Кроме того, по условию

$$\varepsilon_i = 0 \quad \text{для } i = 1, 2, \dots, m.$$

Поэтому, согласно соотношениям (1.20), составляющие квазиплана удовлетворяют условиям (1.2) задачи (A). Следовательно, квазиплан является планом задачи (A). Согласно определению плана $X = (x_1, \dots, x_n)$ при $x_j > 0$ соблюдается равенство

$$\sum_{i=1}^m a_{ij} y_i = c_j.$$

Поэтому имеет место следующая цепочка соотношений:

$$\sum_{j=1}^n c_j x_j = \sum_{j=1}^n x_j \sum_{i=1}^m a_{ij} y_i = \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j = \sum_{i=1}^m b_i y_i.$$

Полученное равенство в силу леммы 5.2 гл. 3 указывает на оптимальность плана X задачи (A). Отсюда следует также, что план Y сопряженной задачи (\bar{A}), породивший вспомогательную задачу (C_Y), является решением задачи (\bar{A}).

Справедливость признака оптимальности доказана.

1.2. Пусть $Y = (y_1, \dots, y_m)$ — некоторый план сопряженной задачи (задачи (\bar{A})). Свяжем с ним вспомогательную задачу (C_Y) и сопряженную вспомогательную задачу (\bar{C}_Y).

Пусть оптимальный план задачи (C_Y) имеет вид

$$(E^*, E^*) = (\xi_{j_1}^*, \dots, \xi_{j_s}^*, e_1^*, \dots, e_m^*),$$

а решением задачи (\tilde{C}_Y) является вектор

$$M^* = (\mu_1^*, \dots, \mu_m^*).$$

Введем параметры Δ_j и δ_j^* , $j = 1, 2, \dots, n$, по формулам

$$\Delta_j = \sum_{i=1}^m a_{ij} y_i - c_j, \quad j = 1, 2, \dots, n, \quad (1.21)$$

$$\delta_j^* = \sum_{i=1}^m a_{ij} \mu_i^*, \quad j = 1, 2, \dots, n. \quad (1.22)$$

Решение вспомогательной задачи (C_Y) определяет квазиплан задачи X . Вектор невязок E^* и параметры δ_j^* являются определяющими характеристиками квазиплана X . В зависимости от значений невязок ε_i^* и знаков параметров δ_j^* следует различать три случая:

1°. Все компоненты вектора невязок равны нулю:

$$\varepsilon_i^* = 0, \quad i = 1, 2, \dots, m.$$

2°. Имеются положительные невязки, и при этом все значения параметров δ_j^* неположительны:

$$\varepsilon_0^* = \sum_{i=1}^m \varepsilon_i^* > 0, \quad \delta_j^* \leq 0 \quad \text{для} \quad j = 1, 2, \dots, n.$$

3°. Вектор невязок содержит положительные составляющие, и среди параметров δ_j^* имеется по крайней мере одна положительная величина.

В случае 1°, как это следует из признака оптимальности, квазиплан оказывается оптимальным планом задачи (A) . Ниже будет показано, что в случае 2° задача (A) неразрешима (условия задачи несовместны), а в случае 3° можно перейти к новому квазиплану с меньшей невязкой.

1.3. Чтобы упростить анализ случаев 2° и 3°, целесообразно проследить за изменением суммы невязок (линейной формы вспомогательной задачи) при переходе от плана Y сопряженной задачи к плану

$$Y(\theta) = Y - \theta M^*.$$

Будем называть переход от плана Y к плану $Y(\theta)$ *элементарным преобразованием* плана Y .

Вычислим параметры $\Delta_j(\theta)$ для плана $Y(\theta)$. В соответствии с формулами (1.21), (1.22) имеем

$$\Delta_j(\theta) = \sum_{i=1}^m a_{ij} y_i(\theta) - c_j = \Delta_j - \theta \delta_j^*, \quad j = 1, 2, \dots, n. \quad (1.23)$$

Вектор $Y(\theta)$ будет планом сопряженной задачи (\tilde{A}) при всех значениях θ , для которых $\Delta_j(\theta) \geq 0$ при $j = 1, 2, \dots, n$.

Вычислим значение линейной формы задачи (\tilde{A}) на плане $Y(\theta)$. Имеем

$$\bar{L}[Y(\theta)] = \sum_{i=1}^m b_i y_i(\theta) = \sum_{i=1}^m b_i y_i - \theta \sum_{i=1}^m b_i \mu_i^*.$$

Применяя первую теорему двойственности к вспомогательной задаче (C_Y) и сопряженной с ней задаче (\tilde{C}_Y), получаем

$$\sum_{i=1}^m b_i \mu_i^* = \sum_{i=1}^m \epsilon_i^*.$$

Следовательно,

$$\tilde{L}[Y(\theta)] = \tilde{L}(Y) - \theta \sum_{i=1}^m \epsilon_i^*. \quad (1.24)$$

Как видим, если $\theta > 0$ и $\epsilon_0^* = \sum_{i=1}^m \epsilon_i^* > 0$, значение линейной формы сопряженной задачи уменьшается при элементарном преобразовании плана Y в план $Y(\theta)$.

Исследуем теперь случай 2°, когда среди невязок ϵ_i^* имеются положительные, а все параметры $\delta_j^* \leq 0$, $j = 1, 2, \dots, n$. В этом случае, как видно из формулы (1.23), вектор $Y(\theta)$ является планом задачи (\tilde{A}) при любом $\theta \geq 0$. Равенство (1.24) показывает, что линейная форма задачи (\tilde{A}) при этом не ограничена снизу на множестве планов этой задачи. Следовательно, согласно лемме 5.3 гл. 3 исходная задача — задача (A) — неразрешима из-за несовместности ее условий. Условия, определяющие случай 2°, будем в дальнейшем называть признаком неразрешимости задачи.

Перейдем к анализу случая 3°, когда имеются положительные невязки ε_i^* и положительные значения параметров δ_j^* .

Наибольшее значение θ , при котором $Y(\theta)$ еще остается планом задачи (\tilde{A}) , определяется соотношением

$$\theta_0 = \min_{\delta_j^* > 0} \left(\frac{\Delta_j}{\delta_j^*} \right). \quad (1.25)$$

Здесь минимум берется по индексам j , для которых величины δ_j^* положительны. В случае 3° такие индексы имеются. Пользуясь определением параметров δ_j^* и условиями (1.17), которым удовлетворяют величины μ_j^* , получаем

$$\delta_j^* \leq 0 \quad \text{при} \quad j \in E_Y.$$

Следовательно, θ_0 достигается при $j \notin E_Y$. С другой стороны, по определению множества E_Y параметры Δ_j положительны при $j \notin E_Y$. Поэтому $\theta_0 > 0$.

Покажем, что план $Y' = Y(\theta_0)$ задачи (\tilde{A}) приводит к вспомогательной задаче $(C_{Y'})$ с меньшим значением минимума суммы невязок, чем для вспомогательной задачи (C_Y) . В силу второй теоремы двойственности при

$$\xi_{j_\lambda}^* > 0$$

имеет место равенство

$$\delta_{j_\lambda}^* = \sum_{i=1}^m a_{ij_\lambda} \mu_i^* = 0.$$

Из полученных соотношений видно, что в $E_{Y'}$ входят, в частности, те индексы j , для которых $x_j = \xi_j^* > 0$. Действительно, для таких j

$$\Delta_j(\theta) = \Delta_j - \theta_0 \delta_j^* = 0 - \theta_0 \cdot 0 = 0.$$

Поэтому исходный план вспомогательной задачи $(C_{Y'})$, отвечающей $Y(\theta_0)$, можно составить из положительных компонент оптимального плана предыдущей вспомогательной задачи (C_Y) . Это обстоятельство существенно сокращает число шагов, необходимых для решения очередной вспомогательной задачи.

Определим индекс j_0 соотношением $\theta_0 = \Delta_{j_0}/\delta_{j_0}^*$ ($\delta_{j_0}^* > 0$). Таких индексов может быть несколько.

Очевидно, вектор A_{j_0} войдет в число векторов условий новой вспомогательной задачи, так как

$$\Delta_{j_0}(\theta_0) = \Delta_{j_0} - \theta_0 \delta_{j_0}^* = 0.$$

Допустим, что задача $(C_{Y'})$ невырожденная (для этого достаточно предположить невырожденность расширенной задачи (B)). В § 2 приводятся соображения, указывающие на целесообразность использования для решения вспомогательной задачи второго алгоритма метода улучшения плана. В соответствии с основными положениями этого метода условие $\delta_{j_0}^* > 0$ для невырожденной задачи означает, что введение в базис исходного плана новой вспомогательной задачи вектора A_{j_0} уменьшит значение ее линейной формы. Поэтому минимум новой вспомогательной задачи строго меньше минимума предыдущей вспомогательной задачи:

$$\sum_{i=1}^m \varepsilon_i^* = \sum_{i=1}^m \varepsilon_i^*(\theta_0) < \sum_{i=1}^m \varepsilon_i^*.$$

Таким образом, в случае 3° можно указать элементарное преобразование плана Y сопряженной задачи в план $Y' = Y(\theta_0)$, который порождает вспомогательную задачу $(C_{Y'})$ с меньшим значением минимума суммы невязок. Другими словами, в случае 3° можно перейти от квазиплана X к квазиплану X' с меньшей невязкой. Описанная последовательность действий составляет одну итерацию метода сокращения невязок. Процесс продолжается до тех пор, пока не придем к случаю 1° или 2°.

1.4. Наметим кратко порядок операций, связанный с отдельным шагом метода последовательного сокращения невязок. В каждой итерации метода предполагается известным (полученным в конце предыдущего шага) план Y сопряженной задачи (\bar{A}) . Итерация начинается с формирования вспомогательной задачи (C_Y) . Решение задачи (C_Y) определяет квазиплан X исходной задачи (A) . Далее производится исследование квазиплана X . По величине невязки квазиплана и знакам параметров δ_j^* выясняется,

какой из случаев (1° , 2° или 3°) имеет место. Если выполняются условия признака оптимальности, т. е. имеет место случай 1° , найденный квазиплан оказывается оптимальным планом задачи (A) и процесс решения завершается. В случае, когда условия признака оптимальности не выполняются, необходимо просмотреть знаки параметров δ_j^* . Неположительность всех δ_j^* указывает на неразрешимость задачи (случай 2°). Если же среди параметров δ_j^* находятся положительные величины, имеет место случай 3° . В этом случае указывается способ построения нового плана сопряженной задачи, отправляясь от которого проводится очередная итерация, обеспечивающая сокращение невязки квазиплана.

Каждая итерация метода сокращения невязок включает несколько шагов метода улучшения плана, необходимых для решения соответствующей вспомогательной задачи. Заметим, что планы любой вспомогательной задачи являются планами расширенной задачи (задачи (B)). Процесс решения вспомогательных задач состоит, таким образом, в последовательных переходах от одного опорного плана расширенной задачи к другому. Движение по опорным планам задачи (B) происходит по правилам метода улучшения плана. Это, однако, не значит, что решение задачи (A) методом сокращения невязок сводится к определению оптимального плана задачи (B) по методу улучшения плана. В самом деле, на протяжении каждой итерации метода сокращения невязок вектор, подлежащий вводу в базис, выбирается не из всей совокупности векторов условий задачи (B), а только из числа единичных векторов e_j ($j = 1, 2, \dots, m$) и тех A_j , для которых

$$\Delta_j = \sum_{i=1}^m a_{ij}y_i - c_j = 0.$$

Здесь $Y = (y_1, \dots, y_m)$ — план задачи (\tilde{A}), связанный с рассматриваемой итерацией. Перед началом каждой итерации множество допустимых векторов A_j обновляется с помощью нового плана задачи (\tilde{A}).

Приведенные соображения позволяют, в частности, сделать вывод о конечности метода сокращения невязок. Действительно, для невырожденной расширенной задачи

переход от одного ее опорного плана к очередному сопровождается монотонным убыванием линейной формы (1.6). Поэтому в процессе решения задачи не может быть возврата к уже пройденному опорному плану задачи (B). Количество различных опорных планов расширенной задачи конечно. Следовательно, получение оптимального плана разрешимой задачи (A) методом сокращения невязок укладывается в конечное число итераций. Неразрешимость задачи, если она имеет место, устанавливается также за конечное число шагов. Если при решении вырожденных вспомогательных задач пользоваться правилами, гарантирующими от заикливания, то вывод о конечности метода сокращения невязок может быть распространен и на случай вырожденности расширенной задачи (B).

1.5. Метод сокращения невязок допускает естественное геометрическое истолкование в $(m+1)$ -мерном пространстве точек $U = (u_1, u_2, \dots, u_m, u_{m+1})$.

Решение задачи (A) методом сокращения невязок начинается с выбора некоторого плана Y сопряженной задачи (\bar{A}). Геометрическим аналогом плана Y является гиперплоскость Π , содержащая начало координат и расположенная «над» конусом K задачи (A). Гиперплоскость Π содержит расширенные векторы \bar{A}_j для $j \in E_Y$. Пересечение гиперплоскости Π и конуса K также является многогранным конусом. Обозначим его через K_Π . Многогранный конус K_Π порожден расширенными векторами \bar{A}_j при $j \in E_Y$. При $m=2$ конус K_Π может либо состоять из одной точки (из начала координат), либо совпадать с ребром конуса K или с плоским углом — двумерным конусом. Размерность K_Π в точности равна m , если Y — опорный план задачи (\bar{A}).

Плану Y задачи (\bar{A}) соответствует вспомогательная задача (C_Y). Конус K_Y задачи (C_Y) определяется векторами условий A_j ($j \in E_Y$) и расширенными единичными векторами

$$\bar{e}_i = (0, \underbrace{\dots, 0, 1}_{i}, 0, \dots, 0, 1), \quad i = 1, 2, \dots, m.$$

Многогранный конус K_Y касается, таким образом, гиперплоскости $u_{m+1}=0$ своей гранью, содержащей векторы A_j ($j \in E_Y$).

Решение задачи (C_Y) сводится к определению «нижней» точки пересечения прямой Q и конуса K_Y . (Во вспомогательной задаче требуется вычислить минимум линейной формы). Ясно, что задача (C_Y) всегда имеет решение. Решив вспомогательную задачу, найдем оптимальные планы взаимосопряженных задач (C_Y) и (\tilde{C}_Y) . Решению задачи (\tilde{C}_Y) соответствует гиперплоскость Π_Y , расположенная «под» конусом K_Y . Обозначим через $M^* = (\mu_1^*, \dots, \mu_m^*, -1)$ направляющий вектор гиперплоскости Π_Y и через Q_Y — «нижнюю» точку пересечения конуса K_Y и прямой Q .

Точка Q_Y является образом решения задачи (C_Y) . Вектор Q_Y может быть представлен в виде

$$Q_Y = \sum_{j \in E_Y} x_j A_j + \sum_{i=1}^m \varepsilon_i^* \bar{e}_i.$$

Рассмотрим на гиперплоскости $u_{m+1}=0$ точку $P_Y = \sum_{j \in E_Y} x_j A_j$

$\left(P_Y = Q_Y - \sum_{i=1}^m \varepsilon_i^* \bar{e}_i \right)$. Проведем через точку P_Y прямую P , параллельную оси Ou_{m+1} . «Верхняя» точка X пересечения прямой P и конуса K задачи (A) является образом квазиплана X .

Назовем «расстоянием» между точками $U' = (u'_1, \dots, u'_{m+1})$ и $U'' = (u''_1, \dots, u''_m, u''_{m+1})$ величину

$$\sum_{i=1}^{m+1} |u''_i - u'_i|.$$

Точка X — образ квазиплана X — обладает следующим экстремальным свойством. Из всех точек $U = (u_1, \dots, u_m, u_{m+1})$, которые

- принадлежат конусу K ,
- принадлежат гиперплоскости Π ,
- удовлетворяют условиям $u_i \leq b_i$,

точка X является ближайшей (в смысле введенного расстояния) к «верхней» точке пересечения конуса K и прямой Q .

В смысле введенного определения «расстояния» длина отрезка $Q_Y B$ (точка B отвечает вектору ограничений) измеряется невязкой $\varepsilon^* = \sum_{i=1}^m \varepsilon_i^*$ квазиплана X .

Возможны два случая: точка Q_Y — образ решения задачи (C_Y) — принадлежит или не принадлежит гиперплоскости $u_{m+1} = 0$.

В первом случае квазиплан X является решением задачи (A) (случай 1°). Действительно, если точка Q_Y совпадает с точкой B , то вектор ограничений принадлежит конусу, порожденному векторами условий A_j ($j \in E_Y$). Следовательно, вектор B является линейной комбинацией векторов A_j ($j \in E_Y$) с неотрицательными коэффициентами. Такой же линейной комбинацией расширенных векторов \bar{A}_j ($j \in E_Y$) является «верхняя» точка пересечения прямой Q и конуса K задачи (A) . План Y в этом случае — решение задачи (\bar{A}) , а X — оптимальный план исходной задачи.

Рассмотрим случай, когда точка Q_Y не принадлежит гиперплоскости $u_{m+1} = 0$. Обозначим через $Y^* = (y_1, y_2, \dots, y_m, -1)$ направляющий вектор гиперплоскости Π . Направляющий вектор гиперплоскости π_Y , отвечающей решению задачи (\bar{C}_Y) , был обозначен ранее через $M^* = (\mu_1^*, \dots, \mu_m^*, -1)$.

Рассмотрим гиперплоскость $\bar{\pi}_Y$, содержащую общую часть гиперплоскостей π_Y и $u_{m+1} = 0$ и параллельную оси Ou_{m+1} . Первые m компонент направляющих векторов \bar{M}^* и M^* гиперплоскостей $\bar{\pi}_Y$ и π_Y совпадают. Последняя компонента вектора \bar{M}^* равна нулю.

В случае, когда точка Q_Y не принадлежит гиперплоскости $u_{m+1} = 0$, могут встретиться две возможности:

а) все векторы \bar{A}_j ($j \notin E_Y$) лежат по ту же сторону от $\bar{\pi}_Y$, что и векторы A_j ($j \in E_Y$). В этом случае гиперплоскость $\bar{\pi}_Y$ разделяет конус K и прямую Q . Это означает, что условия задачи (A) несовместны (случай 2°);

б) по крайней мере один из векторов \bar{A}_j ($j \notin E_Y$) и векторы \bar{A}_j ($j \in E_Y$) лежат по разные стороны от гиперплоскости $\bar{\pi}_Y$. Поворот гиперплоскости Π относительно пересечения S гиперплоскостей Π и $\bar{\pi}_Y$ приводит к захвату

некоторого вектора \bar{A}_j ($j \notin E_Y$). Зафиксированная при этом гиперплоскость Π' отвечает плану Y' сопряженной задачи, с которой связана вспомогательная задача и квазиплан X' с меньшей невязкой. Имеет место случай 3°.

В отличие от ранее рассмотренных методов движение к оптимальному плану методом последовательного сокращения невязок производится по границе конуса K , соответствующего задаче.

§ 2. АЛГОРИТМ МЕТОДА

2.1. Мы уже видели, что каждой итерации метода сокращения невязок отвечает некоторый план Y двойственной задачи (\tilde{A}). План Y в свою очередь определяет вспомогательную задачу (C_Y) данной итерации. Решению вспомогательной задачи соответствует квазиплан X исходной задачи (A). Признак оптимальности — основа метода — позволяет проверить, является ли квазиплан X решением задачи (A). Признак неразрешимости является достаточным условием для отсутствия планов у задачи (A). Если квазиплан X не является оптимальным планом задачи и нет оснований утверждать, что задача неразрешима, следует при помощи элементарного преобразования, определяемого решениями пары двойственных вспомогательных задач (C_Y) и (\tilde{C}_Y), перейти к новому плану Y' сопряженной задачи (\tilde{A}). Отправляясь от плана Y' , повторяют весь процесс — проводят очередную итерацию.

Таким образом, каждый шаг метода сокращения невязок требует решения вспомогательной задачи (C_Y). Вообще говоря, для решения задачи (C_Y) может быть использован как метод улучшения плана, так и метод уточнения оценок. При этом в каждом методе решение задачи может быть получено как по первому, так и по второму алгоритму. Однако имеются соображения, заставляющие рекомендовать для решения вспомогательной задачи второй алгоритм метода улучшения плана. Выбор метода улучшения плана обусловлен тем, что определение исходного плана каждой вспомогательной задачи не требует специальных вычислений. В качестве базиса начального плана первой вспомогательной задачи можно принять

систему единичных векторов e_1, e_2, \dots, e_m . При этом базисные компоненты начального опорного плана

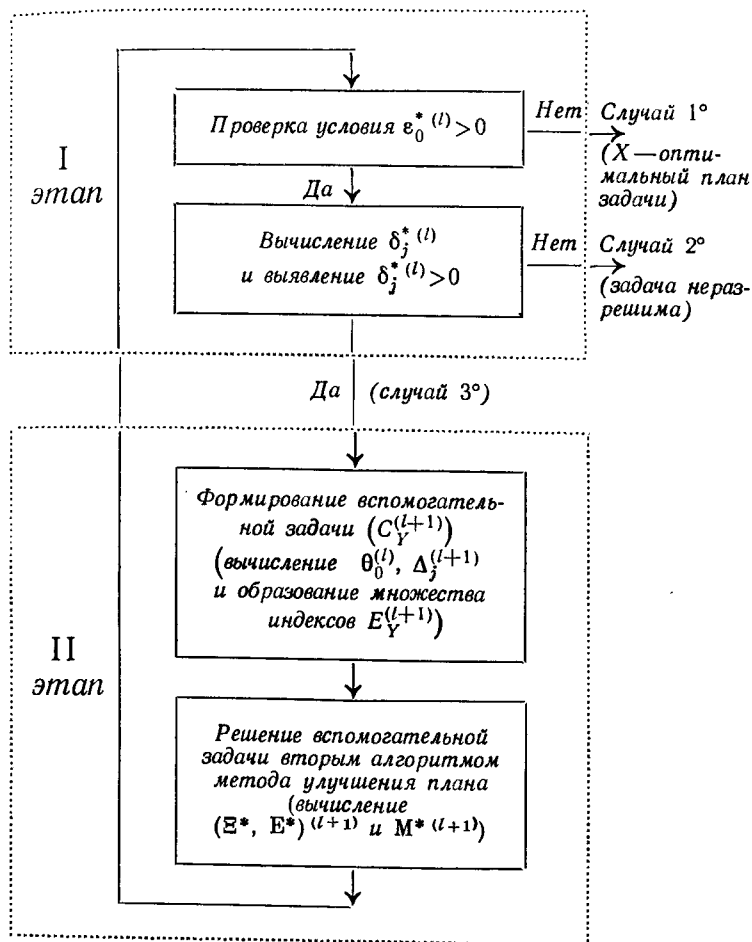


Рис. 6.1.

задачи (C_Y) совпадают с соответствующими составляющими b_i вектора ограничений B . (Величины b_i всегда можно считать неотрицательными.) В качестве исходного плана

каждой последующей вспомогательной задачи можно принимать решение предыдущей.

Для перехода к очередной итерации метода сокращения невязок необходимо иметь не только решение вспомогательной задачи (C_Y), но и оптимальный план сопряженной с ней задачи (\tilde{C}_Y). Поэтому для решения вспомогательной задачи предпочтительнее второй алгоритм метода улучшения плана, в котором наряду с оптимальным планом задачи вычисляется решение сопряженной задачи. Нижняя строка заключительной основной таблицы второго алгоритма заполняется компонентами оптимального плана сопряженной задачи.

Использование первого алгоритма затрудняется еще и тем, что для формирования очередной вспомогательной задачи необходимо получить разложение всех ее векторов условий, которые не входили в предыдущую вспомогательную задачу, по векторам текущего базиса. Во втором алгоритме эта операция не нужна.

Приведенные соображения позволяют положить в основу построения вычислительной схемы метода сокращения невязок второй алгоритм метода улучшения плана.

По аналогии с тем, как это сделано в двух предыдущих главах, можно, используя общую схему метода, построить его вычислительную схему. Мы этого делать не будем. Приведем лишь блок-схему алгоритма метода сокращения невязок (рис. 6.1) и проиллюстрируем вычислительную схему следующим численным примером.

2.2. Требуется обратить в максимум линейную форму

$$L(X) = x_1 + x_2 + x_3 + 3x_4 + x_5 + 3x_6 + x_7$$

при условиях

$$\begin{aligned} x_1 + 2x_2 + 2x_3 + 3x_4 + 2x_5 + 4x_6 + 2x_7 &= 5, \\ 2x_1 + x_2 + 4x_3 - x_4 + x_5 + 2x_6 - 3x_7 &= 7, \\ x_1 + 2x_2 + x_3 + 4x_4 - x_5 + 3x_6 + 2x_7 &= 5, \\ -x_1 + 2x_2 + x_3 + x_4 + 2x_5 - x_6 + x_7 &= 2, \\ x_j &\geq 0, \quad j = 1, 2, \dots, 7. \end{aligned}$$

Все коэффициенты первого ограничения положительны. Согласно рекомендациям § 4 гл. 5 компоненты исходного плана сопряженной задачи можно определить следующим образом:

$$y_1 = \max \left\{ \frac{1}{1}, \frac{1}{2}, \frac{1}{2}, \frac{3}{3}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2} \right\} = 1, \quad y_2 = y_3 = y_4 = 0.$$

Составляем вспомогательную таблицу. В строку Δ записываются величины

$$\Delta_j = \sum_{i=1}^4 a_{ij} y_i - c_j, \quad j=1, 2, \dots, 7.$$

В нуль обращаются два значения Δ (Δ_1 и Δ_4). Следовательно, вспомогательная задача (C_Y) содержит, кроме единичных векторов условий e_i ($i=1, \dots, 4$), только два вектора A_1 и A_4 . Заполняем главную часть начальной основной таблицы. В последней ее строке в позициях столбцов e_1, \dots, e_4 записываются относительные оценки условий задачи (C_Y). Имеем $\mu_i=1$, $i=1, 2, 3, 4$.

Затем вычисляются относительные оценки векторов условий задачи (C_Y). Имеем $\delta_1 = \sum_{i=1}^4 a_{i1} \mu_i = 3$, $\delta_4 = 7$.

Оценки векторов A_1 и A_4 положительны. В очередной базис задачи (C_Y) вводится вектор A_4 с большей оценкой. Коэффициенты разложения вектора A_4 по начальному базису, совпадающие с компонентами A_4 , записываются в столбец A_k основной таблицы. В $(m+1)$ -ю (пятую) позицию столбца A_k помещается величина $\delta_4=7$. Затем для неотрицательных элементов столбца A_k вычисляются составляющие столбца θ — отношения соответствующих элементов столбцов e_0 и A_k . Минимальный элемент $\theta_0=5/4$ столбца θ достигается на векторе базиса e_3 . Вектор e_3 подлежит исключению из базиса.

Продолжая процесс решения вспомогательной задачи по правилам второго алгоритма метода улучшения плана, получаем после двух итераций решения задач (C_Y) и (\bar{C}_Y)

$$(E^*, E^*) = \left(\frac{11}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, \frac{16}{3} \right), \quad M^* = \left(1, \frac{4}{9}, -\frac{8}{9}, 1 \right).$$

Невязка соответствующего квазиплана равна $17/3$. Следовательно, полученный квазиплан не является решением задачи.

Заполняем свободные позиции строки δ^* вспомогательной таблицы. Имеем

$$\delta^* = \left(0, \frac{16}{9}, \frac{35}{9}, 0, \frac{16}{3}, \frac{11}{9}, -\frac{1}{9} \right)$$

В последней строке θ субтаблицы, отвечающей первой итерации, заполняются только четыре позиции (позиции, в которых $\delta_j^* > 0$). Наименьший элемент θ_0 строки θ равен $3/16$.

Переходим ко второй итерации. Элементы первой строки субтаблицы, отвечающей второй итерации, вычисляются по формуле $\Delta'_j = \Delta_j - \theta_0 \delta_j^*$. В строке Δ субтаблицы 2 три нулевых элемента: $\Delta'_1, \Delta'_4, \Delta'_5$. Следовательно, помимо единичных векторов e_i ($i=1, 2, 3, 4$), новая вспомогательная задача ($C_{Y'}$) содержит векторы условий A_1, A_4 и A_5 .

Таблица 6.1 (1—4)

	N	σ	B	e_0	e_1	e_2	e_3	e_4	A_k	θ	N_T
	1	1	e_1	5	1				3	$5/3$	1
	2	1	e_2	7		1			-1	—	
←	3	1	e_3	5			1		4	$5/4$	
	4	1	e_4	2				1	1	2	
	5	μ_0	—	19	1	1	1	1	7	—	
	1	1	e_1	$5/4$	1		$-3/4$		$1/4$	5	1'
←	2	1	e_2	$33/4$		1	$1/4$		$9/4$	$11/3$	
→	3		A_4	$5/4$			$1/4$		$1/4$	5	
	4	1	e_4	$3/4$			$-1/4$	1	$-5/4$	—	
	5	μ'_0	—	$41/4$	1	1	$-3/4$	1	$5/4$	—	
←	1	1	e_1	$1/3$	1	$-1/9$	$-7/9$		$8/3$	$1/8$	1''
→	2		A_1	$11/3$		$4/9$	$1/9$		$1/3$	11	
	3		A_4	$1/3$		$-1/9$	$2/9$		$-1/3$	—	
	4	1	e_4	$16/3$		$5/9$	$-1/9$	1	$8/3$	2	
	5	μ''_0	—	$17/3$	1	$4/9$	$-8/9$	1	$16/3$	—	
↔	1		A_5	$1/8$	$3/8$	$-1/24$	$-7/24$		$7/24$	$3/7$	2
	2		A_1	$29/8$	$-1/8$	$11/24$	$5/24$		$43/24$	$87/43$	
	3		A_4	$3/8$	$1/8$	$-1/8$	$1/8$		$1/8$	3	
	4	1	e_4	5	-1	$2/3$	$2/3$	1	$7/3$	$15/7$	
	5	μ_1	—	5	-1	$2/3$	$2/3$	1	$7/3$	—	
→	1		A_3	$3/7$	$9/7$	$-1/7$	-1		$-4/7$	—	3
	2		A_1	$20/7$	$-17/7$	$5/7$	2		$13/7$	$20/13$	
	3		A_4	$3/7$	$2/7$	$-1/7$			$3/7$	1	
←	4	1	e_4	4	-4	1	3	1	4	1	
	5	μ_2	—	4	-4	1	3	1	4	—	
	1		A_3	1	$5/7$		$-4/7$	$1/7$			4
	2		A_1	1	$-4/7$	$1/4$	$17/28$	$-13/28$			
	3		A_4		$5/7$	$-1/4$	$-9/28$	$-3/28$			
→	4		A_2	1	-1	$1/4$	$3/4$	$1/4$			
	5	μ_3	—		/					—	

Таблица 6.2

N	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Y
1	5	1	2	2	3	2	4	2	1
2	7	2	1	4	-1	1	2	-3	
3	5	1	3	1	4	-1	3	2	
4	2	-1	2	1	1	2	-1	1	
5	C	1	1	1	3	1	3	1	
1	Δ	\times	1	1	\times	1	1	1	
	δ	3			7				
	δ'	$\frac{5}{4}$							
	δ^*		$\frac{16}{9}$	$\frac{35}{9}$		$\frac{16}{3}$	$\frac{11}{9}$	$-\frac{1}{9}$	
	θ	-	$\frac{9}{16}$	$\frac{9}{35}$	-	$\frac{3}{16}$	$\frac{9}{11}$	-	$\theta_0^{(1)} = \frac{3}{16}$
2	Δ	\times	$\frac{2}{3}$	$\frac{13}{48}$	\times	\times	$\frac{37}{48}$	$\frac{49}{48}$	
	δ^*		$\frac{8}{3}$	$\frac{7}{3}$			$-\frac{5}{3}$	$-\frac{5}{3}$	
	θ	-	$\frac{1}{4}$	$\frac{13}{112}$	-	-	-	-	$\theta_0^{(2)} = \frac{13}{112}$
3	Δ	\times	$\frac{5}{14}$	\times	\times	\times	$\frac{27}{28}$	$\frac{17}{14}$	
	δ^*		4			-8	-6	-4	
	θ	-	$\frac{5}{56}$	-	-	-	-	-	$\theta_0^{(3)} = \frac{5}{56}$
4	Δ	\times	\times	\times	\times	$\frac{5}{7}$	$\frac{3}{2}$	$\frac{11}{7}$	

В качестве главной части исходной основной таблицы для решения задачи (C_Y) принимается последняя (третья) основная таблица решения задачи (C_Y) . Вторая итерация метода сокращения невязок состоит в рассматриваемом примере всего из одного шага. Решения третьей и четвертой вспомогательных задач $(C_{Y''})$ и $(C_{Y''''})$ также достигаются за один шаг каждое. Задаче $(C_{Y''''})$ соответствует квази-план

$$X^* = (1, 1, 1, 0, 0, 0)$$

с нулевой невязкой. Квазиплан X^* является оптимальным планом задачи.

Весь ход решения задачи зафиксирован в основной табл. 6.1 и вспомогательной табл. 6.2.

2.3. Использование метода улучшения плана предполагает заданным некоторый опорный план исходной задачи (A) . При решении задачи методом уточнения оценок следует отправляться от некоторого опорного плана сопряженной задачи (\tilde{A}) .

Метод сокращения невязок можно применять, отправляясь от произвольного (не обязательно опорного) плана задачи (\tilde{A}) . Это существенное достоинство метода, поскольку во многих случаях выбор неопорного плана сопряженной задачи не требует специальных вычислений. Если же структура задачи непосредственно не подсказывает вид исходного плана, следует обратиться к методу, изложенному в п. 4.3 гл. 5. В соответствии с этим методом задача (A) дополняется соотношением

$$\sum_{j=1}^n x_j \leq M,$$

или

$$\sum_{j=0}^n x_j = M, \quad x_0 \geq 0.$$

Здесь M предполагается достаточно большим числом. Назовем полученную при этом задачу линейного программирования задач (M) . Сопряженная с ней задача — задача (\tilde{M}) — состоит в отыскании минимума линейной формы

$$\sum_{i=1}^m b_i y_i + M y_0,$$

переменные которой подчинены ограничениям

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

$$y_0 \geq 0.$$

Очевидно, что в качестве плана задачи (\bar{M}) можно принять вектор

$$\bar{Y} = (y_0, 0, \dots, 0),$$

где

$$y_0 = \max_{1 \leq j \leq n} \{c_j, 0\}.$$

Решим теперь задачу (M) методом сокращения невязок, отправляясь от плана \bar{Y} . В процессе решения число M предполагается бóльшим любого числа, с которым его приходится сравнивать. Пусть

$$\bar{X}^* = (x_0^*, x_1^*, \dots, x_n^*)$$

— решение задачи (M) . Возможны два случая:

1°. Индекс $j=0$ принадлежит множеству $E_{Y^{(s)}}$, где $Y^{(s)}$ — план сопряженной задачи, отвечающий последней итерации метода сокращения невязок, т. е. $\Delta_0^{(s)} = 0$.

2°. Индекс $j=0$ не принадлежит множеству $E_{Y^{(s)}}$, т. е. $\Delta_0^{(s)} > 0$.

В первом случае искомым оптимальным планом задачи является вектор

$$X^* = (x_1^*, x_2^*, \dots, x_n^*).$$

Во втором — задача (A) неразрешима. Пользуясь указаниями п. 4.3 гл. 5, читатель сумеет доказать эти утверждения самостоятельно. Нетрудно также доказать, что если процесс решения задачи (M) завершился случаем 2° (задача (M) неразрешима), то условия исходной задачи несовместны.

§ 3. МЕТОД ДВУСТОРОННИХ ОЦЕНОК

3.1. При решении задачи (A) методом сокращения невязок строится последовательность планов сопряженной задачи (\bar{A}) , монотонно сходящаяся к ее решению.

Значение линейной формы сопряженной задачи на планах последовательности, монотонно убывая, стремится к значению линейной формы $L(X^*)$ на оптимальном плане X^* исходной задачи (А). Поэтому после каждой итерации метода легко получить ограничение сверху для оптимальной величины линейной формы $L(X)$ исследуемой задачи (А). Имеются задачи, в которых по каждому квазиплану можно построить план задачи со значением линейной формы, не меньшим, чем на квазиplane. В таких задачах удастся после каждой итерации метода сокращения невязок получить план и оценить разность между значением линейной формы на этом плане и ее оптимальной величиной. Отмеченным полезным свойством обладают, в частности, задачи, в которых легко определяется план при любом векторе ограничений $B \geq 0$, а коэффициенты линейной формы неотрицательны. В этом нетрудно убедиться с помощью следующих рассуждений. Если X' — квазиплан задачи, то $AX' \leq B$. По допущению о структуре матрицы условий A легко определить неотрицательный вектор X'' , удовлетворяющий системе

$$AX'' = B - AX'.$$

В таком случае вектор $X = X' + X''$ является планом исходной задачи, причем в силу неотрицательности коэффициентов линейной формы задачи

$$L(X) \geq L(X').$$

Указанными свойствами обладают, например, задачи, матрицы условий которых содержат полную систему единичных векторов. Таким же свойством обладает и транспортная задача.

3.2. Ниже мы рассмотрим модификацию метода сокращения невязок (ее целесообразно называть *методом двусторонних оценок*), позволяющую в общем случае двигаться по планам исходной задачи. С каждой итерацией метода двусторонних оценок связывается некоторая оценка уклонения полученного плана от оптимального. Решение задачи считается законченным, когда оценка уклонения укладывается в заданные пределы. Указанная возможность заставляет зачастую отдавать предпочтение методу

двусторонних оценок при решении практических задач, в которых может быть задано допустимое отклонение значения линейной формы от оптимального.

Следует, однако, помнить, что для решения задачи линейного программирования методом двусторонних оценок необходимо, кроме начального плана Y сопряженной задачи (\tilde{A}), иметь также начальный опорный план X исходной задачи (A).

Наметим кратко общую схему метода двусторонних оценок. Будем отправляться от опорного плана $X = (x_1, x_2, \dots, x_n)$ задачи (A) и плана $Y = (y_1, y_2, \dots, y_m)$ сопряженной задачи (\tilde{A}). Пусть задача (A) записана в канонической форме. Требуется обратить в максимум линейную форму

$$L(X) = \sum_{j=1}^n c_j x_j \quad (3.1)$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (3.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (3.3)$$

Пусть далее y_1, y_2, \dots, y_m — произвольный заданный набор чисел. Умножая i -е условие системы (3.2) на y_i , суммируя правые и левые части полученных равенств и вычитая из результата соотношение (3.1), получаем

$$\sum_{i=1}^m b_i y_i - L(X) = \sum_{i=1}^m \sum_{j=1}^n (a_{ij} y_i - c_j) x_j.$$

Вводя обозначение

$$\Delta_j = \sum_{i=1}^m a_{ij} y_i - c_j, \quad (3.4)$$

приходим к соотношению

$$\sum_{j=1}^n \Delta_j x_j = \sum_{i=1}^m b_i y_i - L(X). \quad (3.5)$$

Таким образом, задача максимизации $L(X)$ при условиях (3.2) и (3.3) может быть так же сформулирована, как задача минимизации $\sum_{j=1}^n \Delta_j x_j$ при тех же условиях и фиксированных y_i ($i = 1, 2, \dots, m$).

Пусть теперь вектор $Y = (y_1, \dots, y_m)$ — произвольный план задачи (\tilde{A}) , сопряженной с задачей (3.1) — (3.3). Тогда

$$\Delta_j \geq 0, \quad j = 1, 2, \dots, n. \quad (3.6)$$

Если X — план задачи (A) , то значение линейной формы (3.5) может быть использовано для оценки отклонения плана X от оптимального. Действительно,

$$\sum_{j=1}^n c_j x_j \leq L(X_{\text{опт}}) \leq \sum_{i=1}^m b_i y_i.$$

Поэтому процесс решения может быть закончен, когда значение разности $\sum_{i=1}^m b_i y_i - \sum_{j=1}^n c_j x_j = \sum_{j=1}^n \Delta_j x_j$ станет величиной, соизмеримой с погрешностью измерений исходных данных задачи.

Пусть, как и прежде, E_Y представляет собой совокупность индексов j , для которых $\Delta_j = 0$, где Δ_j вычисляются по формуле (3.4), исходя из плана Y сопряженной задачи. Обозначим через E_X множество индексов векторов базиса опорного плана X . Введем, кроме того, множество $E_{X,Y}$, содержащее все индексы j , принадлежащие к E_X или к E_Y ($E_{X,Y} = E_X \cup E_Y$). Свяжем с парой планов X и Y вспомогательную задачу $(D_{X,Y})$, в которой требуется обратить в минимум линейную форму

$$\sum_{j \in E_{X,Y}} \Delta_j x_j \quad (3.7)$$

при условиях

$$\sum_{j \in E_{X,Y}} A_j x_j = B, \quad (3.8)$$

$$x_j \geq 0, \quad j \in E_{X,Y}. \quad (3.9)$$

Решим задачу $(D_{X,Y})$ в соответствии с правилами второго алгоритма метода последовательного улучшения плана. В качестве исходного плана задачи $(D_{X,Y})$ естественно принять заданный опорный план X . Ясно, что каждый план задачи $(D_{X,Y})$ является также планом задачи (A) . (Те компоненты плана, индексы которых не входят в $E_{X,Y}$, полагаются равными нулю.)

Пусть вектор X^* является оптимальным планом задачи $(D_{X, Y})$.¹ Соотношение (3.5) показывает, что значение линейной формы (3.7) на плане X^* естественно назвать *невязкой* ε^* плана X^* задачи (A):

$$\varepsilon^* = \sum_{j \in E_{X, Y}} \Delta_j x_j^*. \quad (3.10)$$

Признак оптимальности плана задачи (A) формулируется следующим образом. План X^* является решением задачи (A), если его невязка равна нулю.

Введем, как и в методе сокращения невязок, параметры

$$\delta_j^* = \sum_{i=1}^m a_{ij} \mu_i^*, \quad j = 1, 2, \dots, n, \quad (3.11)$$

где $M^* = (\mu_1^*, \mu_2^*, \dots, \mu_m^*)$ — решение задачи $(\tilde{D}_{X, Y})$, сопряженной с вспомогательной задачей $(D_{X, Y})$. Решение задачи $(D_{X, Y})$ и сопряженной с ней задачи $(\tilde{D}_{X, Y})$ определяют новый опорный план $X' = X^*$ задачи (A) и план Y' задачи (\tilde{A}) . Невязка плана X' меньше невязки исходного плана X .

Переход от плана Y к плану Y' задачи (\tilde{A}) производится с помощью элементарного преобразования

$$Y' = Y - \theta_0 M^*. \quad (3.12)$$

Параметр θ_0 элементарного преобразования вычисляется по формуле

$$\theta_0 = \min_{\delta_j^* > 0} \frac{\Delta_j}{\delta_j^*}. \quad (3.13)$$

Величины Δ_j' очередной итерации связаны с параметрами предшествующего шага рекуррентными соотношениями

$$\Delta_j' = \Delta_j - \theta_0 \delta_j^*. \quad (3.14)$$

Опорный план $X' = X^*$ задачи (A) и план Y' задачи (\tilde{A}) определяют новую вспомогательную задачу $(D_{X', Y'})$. При этом в качестве исходного плана задачи $(D_{X', Y'})$ естественно принять оптимальный план X^* задачи $(D_{X, Y})$.

Описанный процесс продолжается до получения плана с нулевой невязкой. Задача (A) всегда разрешима, поскольку по условию процесс решения задачи начинается с заданного опорного плана X задачи (A) и плана Y задачи (\tilde{A}) .

3.3. В заключение главы подчеркнем достоинства метода последовательного сокращения невязок, выгодно отличающие его от других методов линейного программирования. Процесс вычисления оптимального плана методом сокращения невязок не требует предварительного определения опорного плана прямой или сопряженной задачи. Для начала процесса достаточно знать лишь произвольный план сопряженной задачи. При наличии опорного плана исходной задачи и произвольного плана сопряженной задачи [модификация метода сокращения невязок — метод двусторонних оценок — позволяет в ряде случаев получить приближенное решение с ошибкой, не превышающей допустимую, при существенно меньшем числе операций, чем это требуется для получения точных значений составляющих оптимального плана.

НЕКАНОНИЧЕСКАЯ ФОРМА ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

До сих пор методы линейного программирования излагались применительно к канонической форме задачи. Единая форма записи задачи упрощает анализ методов и описание вычислительных схем. Это отнюдь не означает, что решение каждой задачи линейного программирования всегда следует начинать с предварительного сведения ее к канонической форме. Часто сохранение естественной формы задачи, определяемой тем или иным приложением, позволяет существенно упростить ее численный анализ. В связи с этим представляет интерес исследование общей задачи линейного программирования, заданной в произвольной форме записи. Этому вопросу посвящен § 1. Здесь предполагается, что в числе ограничений задачи встречаются как равенства, так и неравенства, а условие неотрицательности налагается лишь на часть переменных.

В § 2 рассматриваются некоторые специальные формы записи общей задачи линейного программирования и исследуются условия, при которых целесообразно приводить задачу к канонической форме. В § 3 метод улучшения плана распространяется на задачи линейного программирования, у которых все ограничения имеют вид равенств, а переменные ограничены с обеих сторон. Случай двусторонних ограничений часто возникает в разнообразных приложениях линейного программирования.

Настоящая глава посвящена распространению метода последовательного улучшения плана на задачи линейного программирования, записанные в различной форме. Следуя приведенным рекомендациям, читатель сумеет переписать и другие конечные методы и соответствующие алгоритмы, не связываясь с особенностями канонической формы задачи.

§ 1. ЗАДАЧА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ В ПРОИЗВОЛЬНОЙ ФОРМЕ ЗАПИСИ

1.1. Рассмотрим задачу линейного программирования в наиболее общей форме записи. Требуется вычислить вектор $X = (x_1, x_2, \dots, x_n)$, обращающий в максимум линейную форму

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

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, p, \quad (1.2)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = p+1, p+2, \dots, m, \quad (1.3)$$

$$x_j \geq 0, \quad j = q+1, q+2, \dots, n. \quad (1.4)$$

Естественно предполагать уравнения (1.2) линейно независимыми, так как в противном случае часть из них можно было бы отбросить. Будем, кроме того, считать линейно независимой систему векторов $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$ ($j = 1, 2, \dots, q$). Это условие также не сужает общности дальнейших рассуждений: ему всегда можно удовлетворить за счет уменьшения числа переменных задачи, не подчиняющихся ограничению (1.4). Последнее допущение гарантирует наличие опорных планов у задачи (1.1) — (1.4).

Понятие опорного плана, сформулированное в п. 3.2 гл. 3, может быть следующим образом уточнено применительно к задаче (1.1) — (1.4).

План $X = (x_1, x_2, \dots, x_n)$ задачи (1.1) — (1.4) называется *опорным*, если он порождает неособенную квадратную подматрицу

$$A_X = \begin{vmatrix} a_{i_1 j_1} & a_{i_1 j_2} & \dots & a_{i_1 j_s} \\ a_{i_2 j_1} & a_{i_2 j_2} & \dots & a_{i_2 j_s} \\ \dots & \dots & \dots & \dots \\ a_{i_s j_1} & a_{i_s j_2} & \dots & a_{i_s j_s} \end{vmatrix}$$

матрицы условий $A = \|a_{ij}\|_{m,n}$ такую, что:

а) столбцы матрицы A_X содержат все столбцы матрицы условий A , для которых компонента x_j положительна в плане X или не ограничена требованием неотрицательности, $j_\beta = \beta$ для $\beta = 1, 2, \dots, q$;

б) строки матрицы A_X образованы условиями (1.2) и некоторыми из условий (1.3), причем все такие ограничения-неравенства удовлетворяются планом X как строгие равенства, $i_\alpha = \alpha$ для $\alpha = 1, 2, \dots, p$. Подматрица A_X , порождаемая опорным планом X , называется его *базисом*.

В соответствии с общим определением, приведенным в п. 3.7 гл. 3, опорный план X задачи (1.1) — (1.4) с базисом A_X называется *невыврожденным*, если

$$x_{j_\beta} > 0 \quad \text{при} \quad \beta = q + 1, q + 2, \dots, s,$$

$$\sum_{j=1}^n a_{ij} x_j < b_i \quad \text{при} \quad i \neq i_\alpha \quad (\alpha = 1, 2, \dots, s).$$

Доказательство эквивалентности сформулированных определений опорного плана и невырожденного плана задачи (1.1) — (1.4) и соответствующих понятий гл. 3 предоставляется читателю.

1.2. Опишем метод улучшения плана применительно к невырожденному случаю, когда все опорные планы задачи (1.1) — (1.4) оказываются невырожденными.

Допустим, что нам известен некоторый опорный план X задачи (1.1) — (1.4) и его базис A_X образован элементами, стоящими на пересечении строк и столбцов матрицы A с номерами i_1, i_2, \dots, i_s и j_1, j_2, \dots, j_s соответственно. Процесс решения начинается с проверки плана X на оптимальность. Для этого определяется решение $(\lambda_1, \lambda_2, \dots, \lambda_s)$ системы

$$\sum_{\alpha=1}^s a_{i_\alpha j_\beta} \lambda_\alpha = c_{j_\beta}, \quad \beta = 1, 2, \dots, s, \quad (1.5)$$

и вычисляются величины

$$\Delta_j = \sum_{\alpha=1}^s a_{i_\alpha j} \lambda_\alpha - c_j, \quad j = 1, 2, \dots, n.$$

Если

$$\lambda_\alpha \geq 0 \quad \text{для} \quad \alpha = p + 1, \dots, s \quad (1.6)$$

и

$$\Delta_j \geq 0 \quad \text{для } j = 1, 2, \dots, n, \quad (1.7)$$

то в соответствии с критерием оптимальности для задачи (1.1) — (1.4) (теорема 7.2 гл. 3) план X оказывается оптимальным. Назовем эту возможность случаем 1°. Если же хотя бы одно из соотношений (1.6), (1.7) нарушено, то осуществляется элементарное преобразование плана X , в результате которого либо обнаруживается неразрешимость задачи (случай 2°), либо строится улучшенный опорный план X' (случай 3°).

1.3. Напомним геометрический смысл элементарного преобразования метода улучшения плана (см. п. 3.1 гл. 4). Опорный план соответствует вершине многогранного множества задачи. Выберем одну из гиперплоскостей, проходящих через эту вершину и отвечающих условиям типа неравенств, и рассмотрим ребро многогранного множества задачи, которое расположено на пересечении оставшихся гиперплоскостей. Элементарное преобразование, связанное с выбранной гиперплоскостью, состоит в движении по указанному ребру. Если задача приведена к канонической форме, то элементарное преобразование ее опорного плана всегда связано с одной из координатных гиперплоскостей.

В рассматриваемом случае элементарное преобразование может быть порождено как гиперплоскостью вида $x_j = 0$ ($j = q + 1, \dots, n$), так и гиперплоскостью

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

отвечающей соответствующему ограничению системы (1.3). Поэтому при анализе элементарного преобразования опорного плана задачи (1.1) — (1.4) целесообразно различать два случая.

1. Элементарное преобразование определяется координатной гиперплоскостью $x_k = 0$ ($k \neq j_\beta$, $\beta = 1, 2, \dots, s$) (элементарное преобразование первого типа). В результате этого преобразования опорный план X переходит в

$$X(\theta) = X - \theta H = (x_1(\theta), x_2(\theta), \dots, x_n(\theta)),$$

где вектор $H = (h_1, h_2, \dots, h_n)$ определяется соотношением

$$h_j = \begin{cases} x_{\beta k}, & \text{если } j = j_\beta, \beta = 1, 2, \dots, s, \\ -1, & \text{если } j = k, \\ 0 & \text{в остальных случаях.} \end{cases} \quad (1.8)$$

Здесь величины $x_{\beta k}$ вычисляются из системы уравнений

$$\sum_{\beta=1}^s a_{i_\alpha j_\beta} x_{\beta k} = a_{i_\alpha k}, \quad \alpha = 1, 2, \dots, s. \quad (1.9)$$

Из формул (1.8) и (1.9) вытекает, что

$$\sum_{j=1}^n a_{i_\alpha j} x_j(\theta) = b_{i_\alpha} \quad \text{для } \alpha = 1, 2, \dots, s$$

при любом значении θ .

Выпишем условия, налагаемые на θ , при которых $X(\theta)$ является планом задачи, т. е. удовлетворяет оставшимся условиям (1.3) и ограничениям (1.4). Поскольку $x_k(\theta) = x_k + \theta$, параметр θ должен быть неотрицательным. Необходимость соблюдения остальных ограничений системы (1.4) приводит к неравенствам

$$x_{j_\beta} - \theta x_{\beta k} \geq 0, \quad \beta = q+1, \dots, s.$$

Итак, для неотрицательности всех составляющих вектора $X(\theta)$, начиная с $(q+1)$ -й, необходимо и достаточно, чтобы выполнялось условие

$$0 \leq \theta \leq \theta'_0, \quad (1.10)$$

где

$$\theta'_0 = \min_{\substack{x_{\beta k} > 0 \\ \beta > q}} \frac{x_{j_\beta}}{x_{\beta k}}. \quad (1.11)$$

Если $x_{\beta k} \leq 0$ для $\beta = q+1, \dots, s$, то вектор $X(\theta)$ имеет неотрицательные компоненты $x_j(\theta)$ ($j = q+1, \dots, n$) при любом неотрицательном значении θ . В этом случае полагаем $\theta'_0 = \infty$.

Определим величины

$$\left. \begin{aligned} \Delta^{(i)} &= b_i - \sum_{j=1}^n a_{ij} x_j, \quad i = 1, 2, \dots, m, \\ \delta_k^{(i)} &= \sum_{\beta=1}^s a_{i j_{\beta}} x_{\beta k} - a_{ik}, \quad i = 1, 2, \dots, m. \end{aligned} \right\} \quad (1.12)$$

Учитывая (1.8) и (1.12), имеем

$$\Delta^{(i)}(\theta) = b_i - \sum_{j=1}^n a_{ij} x_j(\theta) = \Delta^{(i)} + \theta \delta_k^{(i)}.$$

Поэтому условия (1.3), которым должен удовлетворять вектор $X(\theta)$, переписываются в виде

$$\Delta^{(i)} + \theta \delta_k^{(i)} \geq 0, \quad i = p+1, \dots, m. \quad (1.13)$$

Разрешая систему неравенств (1.13) относительно $\theta \geq 0$ получаем

$$\theta \leq \theta_0', \quad (1.14)$$

где

$$\theta_0' = \min_{\delta_k^{(i)} < 0} \left(-\frac{\Delta^{(i)}}{\delta_k^{(i)}} \right). \quad (1.15)$$

Если $\delta_k^{(i)} \geq 0$ для всех значений i , то полагаем $\theta_0' = \infty$. Неравенства (1.10) и (1.14) определяют совокупность значений θ , при которых $X(\theta)$ является планом задачи (1.1)–(1.4):

$$0 \leq \theta \leq \theta_0',$$

где

$$\theta_0 = \min \{ \theta_0', \theta_0'' \}, \quad (1.16)$$

а θ_0' и θ_0'' определяются из (1.11) и (1.15) соответственно.

Вычислим приращение линейной формы (1.1), связанное с переходом от X к $X(\theta)$:

$$\sum_{j=1}^n c_j x_j(\theta) = \sum_{j=1}^n c_j x_j - \theta \left(\sum_{\beta=1}^s c_{j_{\beta}} x_{\beta k} - c_k \right).$$

Используя системы равенств (1.5) и (1.9), имеем

$$\begin{aligned}\sum_{\beta=1}^s c_{j\beta} x_{\beta k} &= \sum_{\beta=1}^s \left(\sum_{\alpha=1}^s a_{i_{\alpha} j_{\beta}} \lambda_{\alpha} \right) x_{\beta k} = \\ &= \sum_{\alpha=1}^s \left(\sum_{\beta=1}^s a_{i_{\alpha} j_{\beta}} x_{\beta k} \right) \lambda_{\alpha} = \sum_{\alpha=1}^k a_{i_{\alpha} k} \lambda_{\alpha}.\end{aligned}$$

Следовательно,

$$L(X(\theta)) = \sum_{j=1}^n c_j x_j(\theta) = L(X) - \theta \Delta_k. \quad (1.17)$$

2. Элементарное преобразование определяется гиперплоскостью

$$\sum_{j=1}^n a_{i_t j} x_j = b_{i_t} \quad (t = p+1, p+2, \dots, s)$$

(элементарное преобразование второго типа).

В результате этого преобразования план X переходит в

$$X(\theta) = (x_1(\theta), \dots, x_n(\theta)) = X - \theta H,$$

где вектор $H = (h_1, h_2, \dots, h_n)$ определяется в соответствии с формулой

$$h_j = \begin{cases} e_{\beta t}, & \text{если } j = j_{\beta} \ (\beta = 1, 2, \dots, s), \\ 0 & \text{в остальных случаях.} \end{cases} \quad (1.18)$$

Величины $e_{\beta t}$ удовлетворяют системе уравнений

$$\sum_{\beta=1}^s a_{i_{\alpha} j_{\beta}} e_{\beta t} = \begin{cases} 0 & \text{для } \alpha \neq t, \\ 1 & \text{для } \alpha = t. \end{cases} \quad (1.19)$$

Найдем точные границы изменения θ , в пределах которых $X(\theta)$ является планом задачи (1.1) — (1.4). В соответствии с (1.19)

$$\begin{aligned}\sum_{j=1}^n a_{i_{\alpha} j} x_j(\theta) &= \sum_{j=1}^n a_{i_{\alpha} j} x_j - \theta \cdot \sum_{\beta=1}^s a_{i_{\alpha} j_{\beta}} e_{\beta t} = \\ &= \begin{cases} b_{i_{\alpha}}, & \text{если } \alpha \neq t, \\ b_{i_{\alpha}} - \theta, & \text{если } \alpha = t. \end{cases} \end{aligned} \quad (1.20)$$

Поэтому для выполнения условий (1.2), (1.3) при $i = i_\alpha$ ($\alpha = 1, 2, \dots, s$) необходимо и достаточно, чтобы θ было неотрицательным числом.

Пусть по определению

$$\bar{\delta}_i^{(i)} = \sum_{j=1}^n a_{ij} h_j = \sum_{\beta=1}^s a_{ij_\beta} e_{\beta t}, \quad i = 1, 2, \dots, m.$$

В таком случае

$$\Delta^{(i)}(\theta) = b_i - \sum_{j=1}^n a_{ij} x_j(\theta) = \Delta^{(i)} + \theta \bar{\delta}_i^{(i)}.$$

Учитывая неотрицательность θ и условия (1.3), которым должен удовлетворять вектор $X(\theta)$, получаем

$$\theta \leq \theta_0'',$$

где

$$\theta_0'' = \min_{\bar{\delta}_i^{(i)} < 0} \left(-\frac{\Delta^{(i)}}{\bar{\delta}_i^{(i)}} \right). \quad (1.21)$$

Если $\bar{\delta}_i^{(i)} \geq 0$ для всех i , то $\theta_0'' = \infty$.

По определению вектора $X(\theta)$

$$x_j(\theta) = 0 \quad \text{при} \quad j \neq j_\beta, \quad \beta = 1, 2, \dots, s.$$

Следовательно, для соблюдения условий (1.4) необходимо и достаточно, чтобы

$$x_{j_\beta}(\theta) = x_{j_\beta} - \theta e_{\beta t} \geq 0 \quad \text{для} \quad \beta = q+1, \dots, s,$$

или, учитывая неотрицательность θ ,

$$\theta \leq \theta_0',$$

где

$$\theta_0' = \min_{\substack{e_{\beta t} > 0 \\ \beta > q}} \frac{x_{j_\beta}}{e_{\beta t}}. \quad (1.22)$$

В случае, когда $e_{\beta t} \leq 0$ для $\beta = q+1, \dots, s$, полагаем $\theta_0' = \infty$. Итак, искомая область изменения θ определяется соотношением

$$0 \leq \theta \leq \theta_0,$$

где

$$\theta_0 = \min(\theta_0', \theta_0''), \quad (1.23)$$

а θ'_0 и θ''_0 вычисляются по формулам (1.22) и (1.21) соответственно.

При данном элементарном преобразовании второго типа линейная форма (2.1) принимает значение

$$L(X(\theta)) = L(X) - \theta \sum_{j=1}^n c_j h_j.$$

Используя последовательно соотношения (1.18), (1.5) и (1.19), имеем

$$\begin{aligned} \sum_{j=1}^n c_j h_j &= \sum_{\beta=1}^s c_{j_\beta} e_{\beta t} = \sum_{\beta=1}^s \left(\sum_{\alpha=1}^s a_{i_\alpha j_\beta} \lambda_\alpha \right) e_{\beta t} = \\ &= \sum_{\alpha=1}^s \left(\sum_{\beta=1}^s a_{i_\alpha j_\beta} e_{\beta t} \right) \lambda_\alpha = \lambda_t. \end{aligned}$$

Следовательно,

$$L(X(\theta)) = L(X) - \theta \lambda_t. \quad (1.24)$$

1.4. Перейдем к анализу случаев 2° и 3°. Допустим, что среди чисел λ_α ($\alpha = p+1, \dots, s$) и Δ_j ($j = 1, 2, \dots, n$) имеются отрицательные величины. Выберем одну из них (например, наибольшую по абсолютной величине). Дальнейшее течение процесса решения определяется тем, является ли выбранный параметр одним из чисел Δ_j или одним из чисел λ_α . Остановимся на каждом из отмеченных случаев.

1. Выбор пал на параметр $\Delta_k < 0$. В этом случае используется элементарное преобразование первого типа, определяемое гиперплоскостью $x_k = 0$. Вычислим по формулам (1.11), (1.15) и (1.16) число θ_0 . Может оказаться, что $\theta_0 = \infty$. В соответствии со сказанным ранее это эквивалентно выполнению условий

$$\left. \begin{aligned} x_{\beta k} &\leq 0 \quad \text{для} \quad \beta = q+1, \dots, s, \\ \delta_k^{(i)} &\geq 0 \quad \text{для} \quad i = 1, 2, \dots, m. \end{aligned} \right\} \quad (1.25)$$

Если $\theta_0 = \infty$, то $X(\theta)$ является планом задачи (1.1)–(1.4) при любом $\theta \geq 0$. Но, согласно равенству (1.17),

$$\lim_{\theta \rightarrow \infty} L(X(\theta)) = \infty.$$

Таким образом, при соблюдении условий (1.25) (случай 2°) исследуемая задача оказывается неразрешимой. Пусть теперь условия (1.25) не выполняются (случай 3°) и, следовательно, $\theta_0 < \infty$.

Положим

$$X' = X(\theta_0).$$

Проверим, что X' является опорным планом задачи.

а) Пусть вначале $\theta_0 = \theta'_0$. Это означает, что

$$\theta_0 = \frac{x_{jr}}{x_{rk}} \quad (x_{rk} > 0)$$

при некотором $r \geq q+1$. Поскольку $x_{rk} \neq 0$, то из теоремы 2.1 гл. 4 вытекает неособенность матрицы $A_{X'}$, которая получена из базиса A_X плана X путем замены его r -го столбца столбцом $(a_{i_1k}, a_{i_2k}, \dots, a_{i_s k})^T$.

Следовательно, план X' является опорным и его базис представляет собой квадратную матрицу порядка s .

б) Рассмотрим теперь другую возможность, которая может представиться при вычислении θ_0 :

$$\theta_0 = \theta''_0.$$

В этом случае

$$\theta_0 = -\frac{\Delta^{(l)}}{\delta_k^{(l)}} \quad (\delta_k^{(l)} < 0)$$

при некотором $l \neq i_\alpha$ ($\alpha = 1, 2, \dots, s$). Очевидно,

$$\sum_{j=1}^n a_{lj} x'_j = b_l,$$

$$x'_j = 0, \text{ если } j \neq j_\beta \quad (\beta = 1, 2, \dots, s) \text{ и } j \neq k.$$

Поэтому, убедившись в неособенности матрицы

$$A_{X'} = \begin{vmatrix} a_{i_1 j_1} & a_{i_1 j_2} & \dots & a_{i_1 j_s} & a_{i_1 k} \\ a_{i_2 j_1} & a_{i_2 j_2} & \dots & a_{i_2 j_s} & a_{i_2 k} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i_s j_1} & a_{i_s j_2} & \dots & a_{i_s j_s} & a_{i_s k} \\ a_{lj_1} & a_{lj_2} & \dots & a_{lj_s} & a_{lk} \end{vmatrix},$$

мы докажем опорность плана X' . Докажем этот факт способом от противного. Если столбцы матрицы $A_{X'}$

линейно зависимы, то ее последний столбец является линейной комбинацией первых s столбцов, так как эти столбцы по условию линейно независимы. В соответствии с (1.9) коэффициентами линейной комбинации оказываются числа $x_{\beta k}$. Но, по предположению,

$$\delta_k^{(l)} = \sum_{\beta=1}^s a_{l j_{\beta}} x_{\beta k} - a_{lk} \neq 0.$$

Следовательно, допущение о линейной зависимости столбцов матрицы $A_{X'}$ ошибочно; матрица $A_{X'}$ — неособенная. Итак, план X' является опорным, причем его базис $A_{X'}$ в данном случае представляет собой матрицу порядка $s+1$, полученную в результате окаймления матрицы A_X строкой $(a_{lj_1}, a_{lj_2}, \dots, a_{lj_s}, a_{lk})$ и столбцом $(a_{i_1 k}, a_{i_2 k}, \dots, a_{i_s k}, a_{lk})^T$.

По предположению, X — невырожденный опорный план задачи (1.1) — (1.4). Отсюда $x_{i_r} > 0$, $\Delta^{(l)} > 0$ и, следовательно, $\theta_0 > 0$. Обратившись к соотношению (1.17), обнаруживаем, что

$$L(X') = L(X) - \theta_0 \Delta_k > L(X).$$

2. Рассмотрим теперь другой случай, возникающий при выборе параметра $\lambda_t < 0$. В этом случае для улучшения плана используется элементарное преобразование второго типа, определяемое гиперплоскостью

$$\sum_{j=1}^n a_{ijt} x_j = b_{it}.$$

Вычисляем число θ_0 по формулам (1.21) — (1.23). Если $\theta_0 = \infty$, или, что то же самое, если оказываются выполненными условия

$$\left. \begin{aligned} h_{j_{\beta}} &= e_{\beta t} \leq 0 && \text{для } \beta = q+1, \dots, s, \\ \bar{\delta}_t^{(i)} &= \sum_{\beta=1}^s a_{ij_{\beta}} e_{\beta t} \geq 0 && \text{для } i = 1, 2, \dots, m, \end{aligned} \right\} \quad (1.26)$$

то $X(\theta)$ — план задачи (1.1) — (1.4) при любом $\theta \geq 0$. В соответствии с равенством (1.24) это означает неразрешимость исследуемой задачи (случай 2°).

Если условия (1.26) не выполняются, то $\theta_0 < \infty$, что дает возможность определить план $X' = X(\theta)$ (случай 3°). Как и в рассмотренном ранее случае, когда использовалось элементарное преобразование первого типа, план X' оказывается опорным. Для того чтобы в этом убедиться, необходимо снова рассмотреть две возможности, возникающие при вычислении θ_0 .

а) Пусть $\theta_0 = \theta'_0$. Тогда

$$\theta_0 = \frac{x_{jr}}{e_{rt}} \quad (e_{rt} > 0)$$

при некотором $r \geq q+1$. В этом случае

$$x'_j = 0 \quad \text{при } j \neq j_\beta \quad (\beta = 1, 2, \dots, r-1, r+1, \dots, s),$$

$$\sum_{j=1}^n a_{i_\alpha j} x'_j = b_{i_\alpha} \quad \text{при } \alpha = 1, 2, \dots, t-1, t+1, \dots, s.$$

Убедимся в неособенности квадратной матрицы $(s-1)$ -го порядка

$$A_{X'} = \begin{vmatrix} a_{i_1 j_1} & \dots & a_{i_1 j_{r-1}} & a_{i_1 j_{r+1}} & \dots & a_{i_1 j_s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i_{t-1} j_1} & \dots & a_{i_{t-1} j_{r-1}} & a_{i_{t-1} j_{r+1}} & \dots & a_{i_{t-1} j_s} \\ a_{i_{t+1} j_1} & \dots & a_{i_{t+1} j_{r-1}} & a_{i_{t+1} j_{r+1}} & \dots & a_{i_{t+1} j_s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i_s j_1} & \dots & a_{i_s j_{r-1}} & a_{i_s j_{r+1}} & \dots & a_{i_s j_s} \end{vmatrix}.$$

Действительно, определитель матрицы $A_{X'}$ с точностью до знака совпадает с алгебраическим дополнением элемента $a_{i_t j_r}$ матрицы A_X . Поэтому

$$|A_{X'}| = \pm e_{rt} |A_X| \neq 0, \quad \text{так как } e_{rt} \neq 0.$$

Таким образом, X' является опорным планом с базисом $A_{X'}$, который представляет собой матрицу $(s-1)$ -го порядка, образующуюся из A_X вычеркиванием t -й строки и r -го столбца.

б) Допустим другую возможность: $\theta_0 = \theta''_0$. В этом случае

$$\theta_0 = -\frac{\Delta^{(t)}}{\delta_t^{(t)}} \quad (\bar{\delta}_t^{(t)} < 0)$$

при некотором $l \neq i_\alpha$ ($\alpha = 1, 2, \dots, s$). Очевидно, $x'_j = 0$ при $j \neq j_\beta$ ($\beta = 1, \dots, s$),

$$\sum_{j=1}^n a_{ij} x'_j = b_i \text{ при } i = i_\alpha \text{ } (\alpha = 1, 2, \dots, t-1, t+1, \dots, s) \\ \text{и } i = l.$$

Установим неособенность матрицы $A_{X'}$, образованной из A_X заменой ее t -й строки на строку $(a_{lj_1}, a_{lj_2}, \dots, a_{lj_s})$.

В соответствии с (1.19) вектор $(e_{1t}, e_{2t}, \dots, e_{st})^T$ является t -м столбцом матрицы A_X^{-1} . Поэтому коэффициент при t -й строке в разложении вектора $(a_{lj_1}, a_{lj_2}, \dots, a_{lj_s})$ по строкам матрицы A_X равен

$$\bar{\delta}_t^{(l)} = \sum_{\beta=1}^s a_{lj_\beta} e_{\beta t} \neq 0.$$

Применяя далее теорему 2.1 гл. 4 к вектору $(a_{lj_1}, \dots, a_{lj_s})$ и строкам матрицы A_X , приходим к выводу о неособенности матрицы $A_{X'}$. Итак, в данном случае X' оказывается опорным планом с базисом $A_{X'}$.

Из предположения о невырожденности плана X вытекает, что $\theta_0 > 0$. Следовательно, в соответствии с равенством (1.24)

$$L(X') = L(X) - \lambda_t \theta_0 > L(X).$$

Приведенные построения полностью определяют принципы построения вычислительной схемы метода последовательного улучшения плана применительно к общей задаче линейного программирования в произвольной форме записи. Подробное изложение соответствующих алгоритмов см. в [51].

§ 2. ЕСТЕСТВЕННАЯ И КАНОНИЧЕСКАЯ ФОРМЫ ЗАДАЧИ

2.1. Для приведения задачи (1.1) — (1.4) к каноническому виду необходимо ввести $m - p$ новых неотрицательных переменных (по числу неравенств системы (1.3)) и освободиться от q старых переменных, не ограниченных требованием неотрицательности. В результате образуется

задача с $n + m - (p + q)$ неотрицательными переменными и $m - q$ условиями-равенствами. Поэтому при решении приведенной задачи вторым алгоритмом метода улучшения плана необходимо иметь дело с обратными матрицами порядка $\varphi_2 = m - q$. Порядок обратных матриц, с которыми приходится оперировать при решении задачи описанным здесь методом, не превышает $\varphi_1 = \min(m, n)$.

При реализации указанных методов на универсальных цифровых машинах обратная матрица, преобразуемая от шага к шагу, обычно помещается в оперативную память. Поэтому размер этой матрицы является основным параметром, в зависимости от значения которого задача «влезает» или «не влезает» в машину. При ручном счете также выгоднее, чтобы преобразуемые матрицы имели возможно меньший порядок. Отсюда следует, что ответ на вопрос о том, приводить ли задачу (1.1) — (1.4) к каноническому виду или решать описанным здесь методом, зависит от соотношения чисел $\varphi_1 = \min(m, n)$ и $\varphi_2 = m - q$. Если $\varphi_2 \leq \varphi_1$, то предварительное приведение задачи к каноническому виду оправдано. Если же $\varphi_2 > \varphi_1 = n$, то целесообразно применить модификацию метода улучшения плана, описанию которой посвящен этот параграф.

Заметим, что при $p \geq 1$ и $q \geq 1$ размеры задачи (1.1) — (1.4) могут быть сокращены за счет исключения $\tau = \min(p, q)$ переменных (из числа первых q переменных) и отбрасывания τ уравнений системы (1.2). В результате этих преобразований образуется новая задача типа (1.1) — (1.4), причем хотя бы один из ее параметров (p, q) заведомо равен нулю. Однако в процессе сокращения числа переменных матрица условий A задачи изменяется. Поэтому не исключено, что матрица условий A' эквивалентной задачи оказывается менее выгодной для анализа, чем матрица A . Например, матрица A может обладать существенно более высоким процентом нулевых элементов по сравнению с матрицей A' . Естественно, что в подобных случаях указанное преобразование приводит к усложнению вычислительной работы: незначительное уменьшение размеров задачи влечет за собой существенное увеличение трудоемкости отдельной итерации. В тех же случаях, когда матрица A не обладает полезными особенностями и $p, q \geq 1$, исключение части переменных задачи

(1.1) — (1.4) следует признать целесообразным. Приведенные замечания еще раз подчеркивают важность предварительного анализа условий задачи.

2.2. Существует ряд классов задач линейного программирования, которые выгоднее решать в их естественной записи. Отметим некоторые из них.

Рассмотрим общую задачу линейного программирования с n неотрицательными переменными и m условиями-равенствами ($m < n$). Известно, что задача, двойственная по отношению к ней, содержит m переменных, не ограниченных требованием неотрицательности, и n условий-неравенств. Поэтому для двойственной задачи

$$\varphi_1 = \min(m, n) = m, \quad \varphi_2 = n - m.$$

Если $m < n - m$, то решение двойственной задачи целесообразно проводить, не изменяя ее записи. В гл. 5 указывалось, что метод уточнения оценок состоит в применении метода улучшения плана к двойственной задаче. При этом, как нетрудно сообразить, двойственная задача решается в соответствии с методом, описанным в этом параграфе. Поскольку ни одно из переменных двойственной задачи не ограничено условием неотрицательности, в процессе решения используется только элементарное преобразование второго типа и $\theta_0 = \theta_0''$.

Вследствие этого при улучшении плана двойственной задачи (системы предварительных оценок) реализуется только случай, в котором базисы соседних опорных планов различаются одной строкой (или, в терминах исходной задачи, одним вектором условий).

В качестве второго примера приведем постановку общей задачи линейного программирования в форме Л. В. Канторовича [18] (см. гл. 2, § 7). Сформулируем задачу в терминах настоящего параграфа, опустив второстепенные детали.

Производство или группа предприятий должны выпускать m_2 различных конечных продуктов в заранее заданной пропорции (в заданном ассортименте), определяемой числами $\lambda_1, \lambda_2, \dots, \lambda_{m_2}$. При этом используются m_1 производственных факторов, запасы которых составляют b_1, b_2, \dots, b_{m_1} соответствующих единиц. По условию

$b_i \geq 0$ ($i=1, 2, \dots, m_1$). Если $b_i=0$, то это означает, что i -й фактор является полуфабрикатом данного производства. Имеется определенное количество (n) заранее отработанных способов производства. Способ производства с номером j характеризуется вектором

$$\bar{A}_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T,$$

где $m = m_1 + m_2$. Первые m_1 компонент вектора \bar{A}_j определяют затраты соответствующих производственных факторов, связанные с реализацией j -го способа производства в течение единицы времени (с единичной интенсивностью). Остальные m_2 компонент совпадают с количествами соответствующих конечных продуктов, производимых по данному способу в единицу времени. Основная цель производства заключается в получении возможно большего количества конечных продуктов при соблюдении заданных пропорций, с использованием при этом только имеющихся в наличии ресурсов. Под планом производства, как обычно, понимается набор интенсивностей использования различных способов производства:

$$X = (x_1, x_2, \dots, x_n).$$

Задача определения оптимального плана производства имеет следующую математическую формулировку. Требуется определить вектор $X = (x_1, x_2, \dots, x_n, V)$, удовлетворяющий условиям

$$\sum_{j=1}^n a_{m_1+i,j} x_j \geq \lambda_i V, \quad i=1, 2, \dots, m_2, \quad (2.1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i=1, 2, \dots, m_1, \quad (2.2)$$

$$x_j \geq 0, \quad j=1, 2, \dots, n, \quad (2.3)$$

и имеющий максимально возможную $(n+1)$ -ю компоненту

$$L(X) = V. \quad (2.4)$$

Если под комплектом конечных продуктов понимать набор этих продуктов в количествах $\lambda_1, \lambda_2, \dots, \lambda_{m_2}$, то V — число комплектов конечных продуктов, которое можно получить при реализации плана $X = (x_1, x_2, \dots, x_n)$.

Задача (2.1)—(2.4) имеет $n+1$ переменных, из которых n предполагаются неотрицательными, и $m = m_1 + m_2$ условий-неравенств. В данном случае $\varphi_1 = \min(n+1, m_1 + m_2)$, $\varphi_2 = m_1 + m_2 - 1$. Обычно $n \leq m_1 + m_2$ и, следовательно, $\varphi_1 < \varphi_2$. Это означает, что при решении задачи (2.1)—(2.4) целесообразно оставить ее в том естественном виде, в котором она формулируется.

В этой главе мы затронули только метод улучшения плана. Однако это не означает, что другие общие методы линейного программирования могут использоваться лишь для задач, приведенных к каноническому виду. Каждый общий метод линейного программирования может быть приспособлен также и для задач типа (1.1)—(1.4) подобно тому, как это сделано здесь применительно к методу улучшения плана.

§ 3. СЛУЧАЙ ДВУСТОРОННИХ ОГРАНИЧЕНИЙ

3.1. Задачи линейного программирования, в которых все или некоторые из переменных ограничены с обеих сторон, часто встречаются в практике управления и планирования. Поэтому задачи с двусторонними ограничениями заслуживают специального рассмотрения.

Рассмотрим следующую задачу линейного программирования. Требуется обратить в максимум линейную форму

$$L(X) = \sum_{j=1}^n c_j x_j \quad (3.1)$$

при ограничениях

$$\sum_{j=1}^n A_j x_j = B, \quad (3.2)$$

$$\alpha_j \leq x_j \leq \beta_j, \quad j = 1, 2, \dots, n. \quad (3.3)$$

Ранг матрицы $\|a_{ij}\|_{mn} = (A_1, A_2, \dots, A_n)$ предполагается равным m . Задача (3.1)—(3.3) отличается от общей задачи линейного программирования, записанной в канонической форме, лишь условиями (3.3), которые ограничивают ее переменные, вообще говоря, с двух сторон. Мы не будем предполагать, что все переменные задачи ограничены условиями (3.3) с двух сторон; некоторые

из них могут иметь только по одной границе. Если переменная x_j ограничена лишь сверху, $\alpha_j = -\infty$, если — только снизу, $\beta_j = \infty$. В частности, при $\alpha_j = 0$, $\beta_j = \infty$ задача (3.1) — (3.3) превращается в общую задачу линейного программирования, записанную в канонической форме.

Очевидно, задача (3.1) — (3.3) легко приводится к канонической форме. Однако вряд ли этим следует пользоваться при решении задачи. Каноническая форма задачи с двусторонними ограничениями содержит $2n$ переменных и $n + m$ условий-равенств. Следовательно, в процессе решения придется иметь дело с $(n + m)$ -мерными векторами условий. Трудоемкость каждой итерации при этом существенно возрастет.

Простота дополнительных условий (3.3) позволяет предположить, что методы, разработанные для канонической формы задачи, могут быть без существенных усложнений перенесены на задачи с двусторонними ограничениями. Ниже мы убедимся, что это действительно так.

3.2. Введем несколько определений. Векторы $A_j = (a_{1j}, \dots, a_{mj})^T$ назовем *векторами условий* задачи (3.1) — (3.3), а вектор $B = (b_1, \dots, b_m)^T$ — *вектором ограничений* этой задачи.

План $X = (x_1, x_2, \dots, x_n)$ задачи (3.1) — (3.3) назовем *опорным*, если система векторов условий A_j , соответствующих компонентам x_j , для которых

$$\alpha_j < x_j < \beta_j, \quad (3.4)$$

линейно независима. Отметим, что такое определение опорного плана полностью отвечает понятию опорности, введенному в п. 1.1 для произвольной задачи линейного программирования. Доказательство этого факта предоставляем читателю.

Пусть X — произвольный опорный план задачи (3.1) — (3.3). Рассмотрим совокупность всех векторов условий, которым соответствуют компоненты x_j плана X , удовлетворяющие неравенству (3.4). Система m линейно независимых векторов, содержащая указанную совокупность векторов, называется *базисом* опорного плана X . Состав

вляющие опорного плана, отвечающие векторам базиса, будем называть *базисными переменными*, остальные компоненты плана назовем *внебазисными*. Очевидно, внебазисная переменная x_j равна одному из граничных значений соответствующей компоненты плана (α_j или β_j).

3.3. Сформулируем признак оптимальности — основу метода решения задачи с двусторонними ограничениями. Пусть $X = (x_1, \dots, x_n)$ — произвольный опорный план задачи (3.1) — (3.3) с базисом, состоящим из векторов $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Обозначим, как и прежде, множество индексов векторов базиса через I_X . Используя обозначения, принятые в предыдущих главах, имеем

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}, \quad j = 1, 2, \dots, n, \quad (3.5)$$

$$z_j = \sum_{i=1}^m x_{ij} c_{s_i}, \quad j = 1, 2, \dots, n. \quad (3.6)$$

Положим, как и прежде,

$$x_{i0} = x_{s_i}, \quad i = 1, 2, \dots, m,$$

и введем вектор A_0 , удовлетворяющий равенству

$$A_0 = B - \sum_{j \notin I_X} x_j A_j = \sum_{i=1}^m x_{i0} A_{s_i}.$$

Объединяя последнее соотношение с (3.5), получаем

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}, \quad j = 0, 1, 2, \dots, n. \quad (3.5')$$

Напомним, что первый индекс коэффициента разложения x_{ij} (индекс i) указывает номер позиции базиса, в которой расположен вектор A_{s_i} .

Введем далее параметры Δ_j , положив

$$\Delta_j = \begin{cases} z_j - c_j, & \text{если } \alpha_j \leq x_j < \beta_j, \\ c_j - z_j, & \text{если } x_j = \beta_j. \end{cases} \quad (3.7)$$

Очевидно, при $j \in I_X$, в частности при $\alpha_j < x_j < \beta_j$, $\Delta_j = 0$. В дальнейшем мы убедимся, что величины Δ_j являются

аналогами относительных оценок векторов условий задачи, записанной в канонической форме.

Определение (3.7) параметров Δ_j позволяет сформулировать признак оптимальности опорного плана задачи (3.1) — (3.3) в том же виде, что и для задачи, записанной в канонической форме.

Признак оптимальности. *Опорный план X является решением задачи (3.1) — (3.3), если $\Delta_j \geq 0$ для всех $j \notin I_X$.*

Доказательство. Пусть $X' = (x'_1, \dots, x'_n)$ — произвольный опорный план задачи (3.1) — (3.3). Имеем

$$L(X') = \sum_{j=1}^n c_j x'_j = \sum_{j \in I_X} c_j x'_j + \sum_{\alpha} c_j x'_j + \sum_{\beta} c_j x'_j,$$

где под \sum_{α} (\sum_{β}) следует понимать суммирование по индексам $j \notin I_X$, для которых $x_j = \alpha_j$ ($x_j = \beta_j$). Учитывая ограничения (3.3) и условия признака оптимальности, получаем

$$\sum_{\alpha} c_j (x'_j - \alpha_j) \leq \sum_{\alpha} z_j (x'_j - \alpha_j),$$

$$\sum_{\beta} c_j (x'_j - \beta_j) \leq \sum_{\beta} z_j (x'_j - \beta_j).$$

Поэтому

$$\sum_{j \notin I_X} c_j (x'_j - x_j) \leq \sum_{j \notin I_X} z_j (x'_j - x_j).$$

Полученное неравенство позволяет следующим образом оценить значение линейной формы в точке X' :

$$L(X') \leq \sum_{j \in I_X} c_j x'_j + \sum_{j \notin I_X} z_j (x'_j - x_j) + \sum_{j \in I_X} c_j x_j.$$

Учитывая далее формулу (3.6), имеем

$$L(X') \leq \sum_{i=1}^m c_{s_i} [x'_{s_i} + \sum_{j \notin I_X} x_{ij} (x'_j - x_j)] + \sum_{j \in I_X} c_j x_j. \quad (3.8)$$

Покажем, что

$$x'_{s_i} + \sum_{j \notin I_X} x_{ij} (x'_j - x_j) = x_{s_i} = x_{i0} \quad \text{для } i = 1, 2, \dots, m. \quad (3.9)$$

Действительно, X' — план задачи. Поэтому

$$\sum_{i=1}^m A_{s_i} x'_{s_i} = B - \sum_{j \notin I_X} A_j x'_j. \quad (3.10)$$

Умножая обе части равенства (3.5) на $x'_j - x_j$ и складывая результаты для $j \notin I_X$ с соотношением (3.10), получаем

$$\sum_{i=1}^m A_{si} [x'_{si} + \sum_{j \notin I_X} x_{ij} (x'_j - x_j)] = B - \sum_{j \notin I_X} A_j x_j = A_0.$$

Сравним полученное равенство с соотношением (3.5) для $j=0$. Векторы A_{si} ($i=1, 2, \dots, m$) составляют базис опорного плана X . Линейная независимость этой системы векторов убеждает нас в справедливости равенства (3.9). Из формул (3.8) и (3.9) вытекает, что

$$L(X') \leq \sum_{i=1}^m c_{si} x_{i0} + \sum_{j \notin I_X} c_j x_j = \sum_{j=1}^n c_j x_j = L(X). \quad (3.11)$$

Формула (3.11) имеет место для произвольного плана X' . Это означает, что X — оптимальный план задачи (3.1) — (3.3). Справедливость признака оптимальности доказана.

В § 1 гл. 4 было установлено следующее соотношение:

$$z_j = \sum_{i=1}^m \lambda_i a_{ij}, \quad j=1, 2, \dots, n, \quad (3.12)$$

где вектор $\Lambda = (\lambda_1, \dots, \lambda_m)$ определяется из системы уравнений

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j \quad \text{для } j \in I_X. \quad (3.13)$$

Таким образом, параметры z_j можно вычислять как по формулам (3.6), так и из соотношений (3.12). В зависимости от способа вычисления параметров z_j , необходимых для получения оценок Δ_j , будем, как и в § 1 гл. 4, различать две формы признака оптимальности. В первой форме признака величины z_j подсчитываются по формулам (3.6), во второй — из соотношений (3.12). Вторую форму признака оптимальности можно сформулировать следующим образом.

Для оптимальности опорного плана $X = (x_1, \dots, x_n)$ достаточно, чтобы вектор Λ , определенный из системы

(3.13), удовлетворял неравенствам

$$(\Lambda, A_j) = \sum_{i=1}^m a_{ij} \lambda_i \geq c_j, \quad \text{если } x_j = \alpha_j,$$

$$(\Lambda, A_j) = \sum_{i=1}^m a_{ij} \lambda_i \leq c_j, \quad \text{если } x_j = \beta_j.$$

3.4. Для описания общей схемы решения задачи (3.1) — (3.3) целесообразно перенести введенное в § 2 гл. 4 понятие элементарного преобразования на случай, когда переменные ограничены с обеих сторон. В задаче с двусторонними ограничениями элементарное преобразование, связанное с вектором A_j ($j \notin I_X$), вводится по-разному в зависимости от того, с какой из границ (α_j или β_j) совпадает небазисная переменная x_j .

Рассмотрим отдельно оба случая:

а) $x_j = \alpha_j, \quad j \notin I_X;$

б) $x_j = \beta_j, \quad j \notin I_X.$

а) В первом случае с вектором A_j связывается семейство преобразований опорного плана X в план $X(\theta) = (x_1(\theta), \dots, x_n(\theta))$ вида

$$x_\mu(\theta) = \begin{cases} x_{s_i} - \theta x_{ij}, & \text{если } \mu = s_i, \quad i = 1, \dots, m, \\ x_j + \theta, & \text{если } \mu = j, \\ x_\mu, & \text{если } \mu \notin I_X, \quad \mu \neq j. \end{cases} \quad (3.14)$$

Запишем условия, при которых вектор $X(\theta)$ является планом задачи (3.1) — (3.3). Учитывая формулы (3.5), имеем

$$\sum_{j=1}^n x_j(\theta) A_j = \sum_{j=1}^n x_j A_j + \theta [A_j - \sum_{i=1}^m x_{ij} A_{s_i}] = \sum_{j=1}^n x_j A_j = B$$

при любом θ . Следовательно, вектор $X(\theta)$ удовлетворяет условиям (3.2) при любом значении θ .

Определим теперь совокупность значений θ , при которых все компоненты вектора $X(\theta)$ удовлетворяют условиям (3.3). Поскольку $x_j(\theta) = x_j + \theta = \alpha_j + \theta$, то для выполнения неравенств $\alpha_j \leq x_j(\theta) \leq \beta_j$ необходимо и достаточно, чтобы

$$0 \leq \theta \leq \beta_j - \alpha_j = \beta_j - x_j. \quad (3.15)$$

Остальные ограничения на θ определяются условиями

$$\alpha_{s_i} \leq x_{s_i}(\theta) \leq \beta_{s_i} \quad \text{для } i = 1, 2, \dots, m. \quad (3.16)$$

Очевидно, нарушение этих неравенств может наступить лишь при $x_{ij} \neq 0$. Если $x_{ij} > 0$, то

$$x_{s_i}(\theta) = x_{s_i} - \theta x_{ij} \leq x_{s_i} \quad (\theta \geq 0).$$

Следовательно, при $x_{ij} > 0$ для выполнения i -го неравенства системы (3.16) необходимо и достаточно, чтобы

$$\theta \leq \frac{x_{s_i} - \alpha_{s_i}}{x_{ij}}. \quad (3.17)$$

Если $x_{ij} < 0$, то при $\theta > 0$

$$x_{s_i}(\theta) = x_{s_i} - \theta x_{ij} \geq x_{s_i}.$$

Следовательно, при $x_{ij} < 0$ i -е неравенство системы (3.16) будет выполняться, если

$$\theta \leq \frac{\beta_{s_i} - x_{s_i}}{-x_{ij}} = \frac{x_{s_i} - \beta_{s_i}}{x_{ij}}. \quad (3.18)$$

Собирая вместе неравенства (3.15), (3.17) и (3.18), получаем, что вектор $X(\theta)$ удовлетворяет всем условиям (3.3) в том и только в том случае, когда

$$0 \leq \theta \leq \theta_0, \quad (3.19)$$

где θ_0 — наименьшее из трех чисел:

$$\begin{aligned} \theta_0^{(1)} &= \min_{x_{ij} > 0} \frac{x_{s_i} - \alpha_{s_i}}{x_{ij}}, \quad \theta_0^{(2)} = \min_{x_{ij} < 0} \frac{x_{s_i} - \beta_{s_i}}{x_{ij}}, \\ \theta_0^{(3)} &= \beta_j - x_j = \beta_j - \alpha_j. \end{aligned}$$

До сих пор мы рассматривали m позиций базиса. В каждой позиции расположен один из векторов базиса. Для упрощения записи удобно ввести $(m+1)$ -ю позицию базиса, поместив в нее вектор $A_j = A_{s_{m+1}}$, определяющий данное элементарное преобразование. Введение дополнительной позиции будет полезно и в дальнейшем изложении.

Пусть

$$x_{m+1, j} = -1, \quad x_{m+1, 0} = x_j = x_{s_{m+1}}.$$

Если положить

$$\gamma_i = \begin{cases} \alpha_{s_i} & \text{при } x_{ij} > 0, \\ \beta_{s_i} & \text{при } x_{ij} < 0 \end{cases} \quad (i = 1, 2, \dots, m+1), \quad (3.20)$$

то формула для вычисления θ_0 может быть записана в следующей компактной форме:

$$\theta_0 = \min_{\substack{x_{ij} \neq 0 \\ 1 \leq i \leq m+1}} \frac{x_{i0} - \gamma_i}{x_{ij}}. \quad (3.21)$$

Здесь, как обычно, $x_{i0} = x_{s_i}$ — базисная переменная, отвечающая i -й позиции базиса.

Итак, под элементарным преобразованием, связанным с вектором A_j , для которого $x_j = \alpha_j$ ($j \notin I_X$), будем понимать преобразование (3.14) при θ , удовлетворяющем условию (3.19). Величина θ_0 в неравенстве (3.19) определяется по формуле (3.21).

б) Введем теперь понятие элементарного преобразования для второго случая. В этом случае вектору A_j , с которым связано элементарное преобразование, отвечает небазисная переменная $x_j = \beta_j$. Семейство элементарных преобразований плана X в план $X(\theta)$, связанное с вектором A_j , определяется формулой

$$x_\mu(\theta) = \begin{cases} x_{s_i} + \theta x_{ij}, & \text{если } \mu = s_i, \quad i = 1, 2, \dots, m, \\ x_j - \theta, & \text{если } \mu = j, \\ x_\mu, & \text{если } \mu \notin I_X, \quad \mu \neq j. \end{cases} \quad (3.22)$$

Вектор $X(\theta)$ является планом задачи (3.1) — (3.3) в том и только в том случае, если $0 \leq \theta \leq \theta_0$, где θ_0 — наименьшее из трех чисел:

$$\theta_0^{(1)} = \min_{x_{ij} < 0} \frac{\alpha_{s_i} - x_{s_i}}{x_{ij}}, \quad \theta_0^{(2)} = \min_{x_{ij} > 0} \frac{\beta_{s_i} - x_{s_i}}{x_{ij}}, \\ \theta_0^{(3)} = x_j - \alpha_j = \beta_j - \alpha_j.$$

Положим

$$\gamma_i = \begin{cases} \alpha_{s_i}, & \text{если } x_{ij} < 0, \\ \beta_{s_i}, & \text{если } x_{ij} > 0 \end{cases} \quad (i = 1, 2, \dots, m+1). \quad (3.23)$$

В принятых обозначениях получаем следующую компактную

запись формулы для вычисления θ_0 :

$$\theta_0 = \min_{\substack{x_{ij} \neq 0 \\ 1 \leq i \leq m+1}} \frac{y_i - x_{i0}}{x_{ij}}. \quad (3.24)$$

Здесь, как и прежде, $x_{m+1,j} = -1$, $x_{m+1,0} = x_j = x_{s_{m+1}}$. При всех неотрицательных θ , не превышающих θ_0 из (3.24), элементарное преобразование относительно вектора A_j , для которого $x_j = \beta_j$ ($j \notin I_X$), переводит опорный план X в план $X(\theta)$ задачи (3.1) — (3.3). Доказательство этого утверждения полностью совпадает с доказательством аналогичного условия для случая а), когда $x_j = \alpha_j$, и поэтому может быть опущено.

Число θ_0 будем в дальнейшем называть *параметром элементарного преобразования*.

Итак, формулы (3.14) и (3.22) при $0 \leq \theta \leq \theta_0$, где θ_0 вычисляется соответственно из соотношений (3.21) и (3.24), полностью определяют элементарное преобразование, связанное с произвольным вектором условий A_j , $j \notin I_X$.

Проследим теперь за изменением линейной формы в результате элементарного преобразования, связанного с вектором A_j , $j \notin I_X$. Учитывая соотношения (3.14) (для случая а)) и (3.22) (для случая б)), получаем

$$L[X(\theta)] - L(X) = \begin{cases} \theta \left(c_j - \sum_{i=1}^m c_{s_i} x_{ij} \right) & \text{при } x_j = \alpha_j, \\ \theta \left(\sum_{i=1}^m c_{s_i} x_{ij} - c_j \right) & \text{при } x_j = \beta_j. \end{cases}$$

В соответствии с определением оценок Δ_j (см. формулы (3.6) и (3.7)) приходим к следующему результату:

$$L[X(\theta)] - L(X) = -\theta \Delta_j. \quad (3.25)$$

Формула (3.25) показывает, что параметр Δ_j естественно называть оценкой вектора A_j относительно плана X с данным базисом $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ или, короче, относительной оценкой A_j .

Приведенные рассуждения служат основой для построения вычислительной схемы решения задач линейного программирования с двусторонними ограничениями. Соответствующие алгоритмы подробно изложены в [51].

В настоящей главе рассматриваются вопросы, которые представляют интерес для всех конечных методов.

В § 1 приводится классификация конечных методов. Здесь рассматриваются различные признаки, позволяющие установить единый подход ко всем конечным методам линейного программирования.

В § 2 исследуются пути совершенствования конечных методов. Описанная здесь модификация методов (она подробно изложена применительно к методу улучшения плана и намечена для метода уточнения оценок) обеспечивает целесообразное использование самого узкого места современных вычислительных машин — их оперативной памяти.

§ 1. КЛАССИФИКАЦИЯ КОНЕЧНЫХ МЕТОДОВ

1.1. В предыдущих главах подробно изучены три конечных метода линейного программирования: метод последовательного улучшения плана, метод последовательного уточнения оценок и метод последовательного сокращения невязок. Ниже мы убедимся, что имеются основания называть рассмотренные здесь методы основными. В различных главах приводятся замечания о других конечных методах и о возможных модификациях описанных здесь методов. В связи с этим представляется целесообразным рассмотреть различные конечные методы с единой точки зрения, сравнить принципы, на которых они основаны, структуру каждой итерации и порядок перехода от одной итерации к следующей.

Возможны различные подходы к систематизации конечных методов линейного программирования. Методы линейного программирования относят к разным группам

в зависимости от того, используется для достижения решения прямая задача, или сопряженная ей задача, или обе задачи двойственной пары совместно. Задачи линейного программирования имеют наглядное геометрическое и экономическое истолкование. Известные методы решения линейных экстремальных задач могут также классифицироваться с точки зрения каждой из этих интерпретаций. Имеются и другие формальные основания, позволяющие проанализировать конечные методы с единых позиций.

Необходимость единого подхода к различным методам линейного программирования вынуждает нас в следующих пунктах кратко повторить некоторые общие соображения, которые излагались в различных главах применительно к каждому методу в отдельности.

1.2. Естественно различать конечные методы линейного программирования в зависимости от того, получают ли решение задачи при движении по планам прямой задачи, или по планам сопряженной задачи, или при совместном использовании обеих задач двойственной пары.

При таком признаке классификации метод последовательного улучшения плана и различные его модификации следует отнести к конечным методам первой группы, в которых оптимальный план достигается при движении по опорным планам исходной задачи. Процесс решения начинается с анализа исходного опорного плана задачи. В каждой итерации метода осуществляется переход от одного опорного плана задачи к другому, соответствующему, вообще говоря, большему значению линейной формы.

Метод последовательного уточнения оценок и различные его модификации следует отнести ко второй группе конечных методов линейного программирования, в которых решение исходной задачи достигается при движении по планам сопряженной задачи. Процесс решения начинается с анализа заданного опорного плана сопряженной задачи. Каждая итерация метода соответствует переходу от одного опорного плана сопряженной задачи к другому, или, что то же самое, от одного псевдоплана исходной задачи к следующему. Переход к новому плану сопряженной задачи связан, вообще говоря, с уменьшением

ее линейной формы. Иначе, переход к очередному псевдоплану приводит обычно к уменьшению линейной формы исходной задачи. Вычислительная схема метода последовательного уточнения оценок формулируется в терминах исходной задачи.

Метод последовательного сокращения невязок в рассматриваемой классификации относится к третьей группе методов, в которой используются обе задачи двойственной пары. Процесс решения задачи начинается с анализа заданного плана (не обязательно опорного) сопряженной задачи. Каждой итерации метода соответствует переход от одного плана сопряженной задачи к следующему. При этом в расширенной задаче, отличающейся от исходной линейной формой и дополнительными переменными с единичными векторами условий, происходит переход от одного опорного плана к другому. В терминах исходной задачи каждая итерация метода сокращения невязок означает переход от одного квазиплана задачи к следующему, с меньшей невязкой.

Наиболее четко характеристики методов третьей группы выступают в одной из модификаций метода сокращения невязок—в методе двусторонних оценок. Здесь решение задачи начинается с произвольного (не обязательно опорного) плана сопряженной задачи и некоторого опорного плана прямой задачи. В результате каждой итерации производится переход к новому плану сопряженной задачи и новому опорному плану исходной задачи. В методе двусторонних оценок от шага к шагу сокращается невязка плана—разность между значениями линейных форм сопряженной и исходной задач на соответствующих планах.

Приведенные соображения позволяют считать три описанных в настоящей книге метода основными представителями трех групп конечных методов, связанных с принципиально различными подходами к решению задач линейного программирования.

1.3. Ранее были приведены две геометрические интерпретации задачи линейного программирования и сопряженной с ней задачи. Для обзора и сравнения различных конечных методов более удобна вторая интерпретация.

Как обычно, при этом геометрическом истолковании задача предполагается записанной в канонической форме.

В $(m+1)$ -мерном пространстве точек $U = (u_1, \dots, u_{m+1})$ рассматривается выпуклый многогранный конус K и прямая Q . Конус K порожден расширенными векторами условий $\bar{A}_j (j=1, 2, \dots, n)$. Прямая Q проходит через точку $B = (b_1, \dots, b_m, 0)$, определяемую вектором ограничений, параллельно оси Ou_{m+1} . Между $(m+1)$ -мерным пространством точек U и n -мерным пространством точек $X = (x_1, \dots, x_n)$ установлено соответствие, согласно которому конус K отвечает точкам $X \geq 0$, а прямая Q является образом точек с нулевым вектором невязок $B - AX$. (Здесь A — матрица условий задачи, B — вектор ограничений.) Таким образом, общая часть прямой Q и конуса K соответствует точкам X , удовлетворяющим условиям $X \geq 0$ и $AX = B$, и, следовательно, является образом области определения линейной формы задачи.

В разрешимой задаче линейного программирования общая часть прямой Q и конуса K может состоять из отрезка (который может выродиться в точку) или луча. Для определенности будем рассматривать случай, когда пересечение прямой и конуса представляет собой отрезок. Координата u_{m+1} точек U определяет значение линейной формы на соответствующем векторе X . «Верхняя» точка M пересечения прямой Q и конуса K отвечает максимальному значению линейной формы, а «нижняя» точка m — минимальному значению линейной формы в области своего определения. Таким образом, в геометрических терминах задача линейного программирования сводится к отысканию «верхней» (в задачах на максимум) точки пересечения прямой Q и конуса K .

Гиперплоскости, натянутые на m линейно независимых расширенных векторов условий и пересекающие отрезок Mm , соответствуют опорным планам задачи.

Гиперплоскости, натянутые на m линейно независимых расширенных векторов условий и расположенные «над» конусом K , соответствуют опорным планам сопряженной задачи.

Можно доказать, что если образом области определения линейной формы исходной задачи является отрезок

Mt , то области определения линейной формы сопряженной задачи отвечает луч MQ — часть прямой Q , расположенная «над» точкой M (включая эту точку).

Отрезок Mt содержит конечное число точек-образов опорных планов исходной задачи. Точно так же луч MQ содержит конечное число точек, соответствующих опорным планам сопряженной задачи.

Точкам поверхности конуса K , за исключением точек M и t , соответствуют векторы X с ненулевыми векторами невязок. Поверхность конуса может быть разбита на «верхнюю» и «нижнюю» части. Точка U поверхности конуса относится к «верхней» части поверхности, если через луч OU может быть проведена гиперплоскость, разделяющая конус K и положительную полуось Ou_{m+1} . Аналогично определяется «нижняя» часть поверхности конуса.

Построения, описанные в п. 1.5 гл. 6, позволяют выделить конечное число точек «верхней» части поверхности конуса, соответствующих квазипланам задачи максимизации, и точки «нижней» части поверхности конуса — образы квазипланов задачи минимизации.

В приведенных геометрических понятиях легко сформулировать особенности различных конечных методов линейного программирования. Метод последовательного улучшения плана и его модификации соответствуют движению по планам задачи. От итерации к итерации мы переходим от одного опорного плана задачи к другому с, вообще говоря, бóльшим (в задачах максимизации) значением линейной формы. Следовательно, метод улучшения плана соответствует движению по точкам отрезка Mt — образам опорных планов исходной задачи — «вверх» до точки M .

Метод последовательного уточнения оценок и его модификации соответствуют движению по опорным планам сопряженной задачи. Каждая итерация метода определяет переход от одного опорного плана сопряженной задачи к следующему с, вообще говоря, меньшим значением линейной формы. Следовательно, метод уточнения оценок соответствует движению по точкам луча MQ — образам опорных планов сопряженной задачи — «вниз» до точки M .

Каждый шаг метода последовательного сокращения невязок переводит один квазиплан задачи в следующий

с меньшей невязкой. Следовательно, метод сокращения невязок для задач максимизации соответствует движению к оптимуму по точкам «верхней» части поверхности конуса, отвечающим квазипланам задачи.

Возможны конечные методы линейного программирования, которые приводят к оптимуму по более сложному пути. В частности, с точки зрения приведенной геометрической классификации методов метод двусторонних оценок, изложенный в § 3 гл. 6, следует отнести не к модификации метода сокращения невязок, а скорее рассматривать как комбинацию методов улучшения плана и уточнения оценок. В методе двусторонних оценок приближение к оптимуму производится по прямой Q извне и изнутри конуса.

Итак, метод улучшения плана соответствует движению к оптимуму изнутри конуса задачи, метод уточнения оценок — извне конуса, а метод сокращения невязок — по поверхности конуса. Во всех трех случаях движение к оптимуму происходит по точкам, принадлежащим конечному множеству точек.

Таким образом, геометрическая классификация также свидетельствует о том, что три метода, рассмотренные в книге, могут с полным правом называться основными конечными методами линейного программирования.

1.4. Экономическая интерпретация задачи линейного программирования, введенная в § 5 гл. 1, и экономическое истолкование двойственной задачи, изложенное в п. 5.5 гл. 3, позволяют с несколько иной точки зрения подойти к классификации конечных методов.

Как мы уже видели, формулировка задачи в экономических терминах более естественна, если условия задачи записываются в виде неравенств. Прямая задача с линейной формой

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

и условиями

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

$$x_j \geq 0, \quad j=1, 2, \dots, n, \quad (1.3)$$

интерпретируется как задача составления плана работы предприятия (выбора времен или интенсивностей использования технологических способов производства из числа заранее отработанных способов). Искомый план должен обеспечить максимальный выпуск некоторого однородного продукта при заданных ресурсах различных производственных факторов.

В соотношениях (1.1) — (1.3) приняты следующие обозначения: x_j — время, в течение которого предприятие работает по j -му технологическому способу (интенсивность использования j -го способа производства); a_{ij} — количество единиц i -го производственного фактора, используемое в j -м технологическом способе в единицу времени (при единичной интенсивности); b_i — размер запаса i -го производственного фактора; c_j — количество единиц продукции, выпускаемое в единицу времени при использовании одного лишь j -го технологического способа производства.

В некоторых случаях под c_j удобно понимать оценку (цену, стоимость) продукции, выпущенной в единицу времени при j -м способе производства. В зависимости от определения коэффициентов c_j линейная форма означает общий объем или общую оценку (цену, стоимость) выпускаемой продукции.

Каждый технологический способ производства характеризуется расширенным вектором условий \bar{A}_j . Первые m компонент вектора \bar{A}_j определяют вектор затрат, а последняя составляющая совпадает с объемом или оценкой выпускаемой продукции.

Задача планирования производства, принятая за основу экономической интерпретации методов линейного программирования, может быть записана и в канонической форме. Для этого следует ввести m дополнительных фиктивных способов производства с единичными векторами затрат ($A_{m+i} = e_i$) и нулевыми оценками производительности. При способе производства с номером $m+i$ в единицу времени расходуется только i -й производственный фактор (единица i -го фактора в единицу времени) и ничего не производится. Фиктивные технологические способы производства используются в тех планах, которым соответствуют излишки отдельных производственных факторов.

План производства задается вектором X , компоненты которого обозначают времена (интенсивности) использования разных технологических способов. Оптимальный план обеспечивает наибольшую при заданных ресурсах оценку выпускаемой продукции (или, что то же самое, максимальный объем продукции предприятия).

Отправляясь от оценки выпускаемой продукции, можно естественным образом оценить отдельные производственные факторы. Система оценок y_1, y_2, \dots, y_m производственных факторов — план цен — определяется как решение задачи, сопряженной с задачей планирования производства. Эта задача состоит в вычислении минимума линейной формы

$$\bar{L}(Y) = \sum_{i=1}^m b_i y_i \quad (1.4)$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n, \quad (1.5)$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m. \quad (1.6)$$

Линейная форма (1.4) интерпретируется как суммарная оценка ресурсов. Условия (1.5) требуют, чтобы оценка суммарных расходов производственных факторов при каждом технологическом способе производства была не меньше оценки выпускаемой за то же время продукции. Условия (1.6) естественны — оценки не могут быть отрицательными величинами.

План цен определяет оптимальный план производства и, наоборот, оптимальному плану производства соответствует план цен — система оценок производственных факторов. Напомним, что оценки производственных факторов измеряются в единицах ценности выпускаемой продукции.

Ясно, что наиболее экономичными являются способы производства, для которых оценка затрат совпадает с оценкой продукции. Будем называть такие технологические способы рентабельными способами производства. Нерентабельные способы производства убыточны: оценка продукции, произведенной при нерентабельных способах

производства, ниже оценки затрат ресурсов. Нерентабельные технологические способы используют возможности производства не наилучшим способом. Оптимальный план производства составляется из одних только рентабельных технологических способов.

Приведенное экономическое истолкование пары двойственных задач позволяет сформулировать рассмотренные конечные методы в терминах задачи планирования производства.

Процесс решения задачи методом улучшения плана начинается с некоторого опорного плана. В экономической терминологии это значит, что решение задачи начинается с анализа некоторого набора из m неотрицательных чисел $x_{s_1}, x_{s_2}, \dots, x_{s_m}$, определяющих интенсивности (времена) использования каждого из m технологических способов производства с линейно независимыми векторами затрат A_{s_1}, \dots, A_{s_m} . Естественно назвать способы производства, отвечающие векторам базиса, *базисными* технологическими способами.

Разложение вектора затрат A_j по базисным векторам затрат сводится к установлению времен (интенсивностей) x_{ij} использования базисных способов производства, при которых будет израсходовано столько же ресурсов каждого из производственных факторов, сколько расходуется при j -м способе производства в единицу времени. При работе предприятия в соответствии с s_i -м способом производства в течение x_{ij} единиц времени ($i = 1, 2, \dots, m$) использование всех базисных технологических способов

обеспечит выпуск $\sum_{i=1}^m c_{s_i} x_{ij}$ единиц продукции. Оценка

произведенной продукции при этом будет также равна

$\sum_{i=1}^m c_{s_i} x_{ij}$ единицам ценности. С другой стороны, при ис-

пользовании одного лишь j -го способа производства в течение единицы времени выпуск продукции окажется равным c_j единицам продукции (или, что то же самое, оценка продукции будет равна c_j единицам ценности). Таким образом, при одних и тех же затратах производственных факторов объем и, следовательно, оценка производственной продукции в первом случае будет

равна $\sum_{i=1}^m c_{s_i} x_{ij}$, а во втором — c_j . Разность

$$\Delta_j = \sum_{i=1}^m c_{s_i} x_{ij} - c_j$$

является оценкой j -го способа производства по отношению к выбранной системе базисных способов производства. Параметр Δ_j позволяет судить о целесообразности включения в план j -го способа производства за счет соответствующего сокращения интенсивности использования базисных технологических способов. Если $\Delta_j < 0$, такое преобразование целесообразно. При $\Delta_j > 0$ оно привело бы к сокращению выпуска продукции: способ производства с номером j менее экономичен, чем комбинация базисных способов.

Таким образом, анализ исходного плана производства сводится к вычислению параметров Δ_j . Переход к другим базисным способам производства не приведет к увеличению выпуска продукции, если $\Delta_j \geq 0$ для всех j . В этом случае анализируемый план оптимален. Если среди Δ_j имеются отрицательные величины, можно улучшить план производства, заменив один из базисных способов k -м технологическим способом с $\Delta_k < 0$. При этом способ, исключаемый из числа базисных, выбирается так, чтобы новый план производства был реализуем (чтобы $x_j \geq 0$ для всех j). Вновь полученный план снова проверяется на оптимальность. Через конечное число шагов последовательное улучшение плана приведет к оптимальному плану работы предприятия.

Сформулируем теперь в экономических терминах метод последовательного уточнения оценок. Решение задачи начинается с анализа некоторого опорного плана сопряженной задачи. План сопряженной задачи может быть интерпретирован как система предварительных оценок производственных факторов. Система предварительных оценок должна удовлетворять условиям (1.5), (1.6). Опорному плану сопряженной задачи соответствуют, кроме того, m рентабельных (относительно этого плана) способов производства с линейно независимыми векторами затрат. Будем называть эти технологические способы *базисными* способами. Пусть номера базисных способов

производства — s_1, s_2, \dots, s_m . План производства не всегда может быть реализован, если ограничиваться только способами производства, рентабельными относительно заданной системы предварительных оценок. Времена x_{i0} использования рентабельных способов производства должны удовлетворять условиям

$$b_\lambda = \sum_{i=1}^m a_{\lambda s_i} x_{i0}, \quad \lambda = 1, 2, \dots, m. \quad (1.7)$$

Если все x_{i0} , удовлетворяющие системе (1.7), неотрицательны, то исходная система предварительных оценок оказывается системой оценок производственных факторов, т. е. «планом цен», а соответствующий план производства оптимален. Наличие отрицательных компонент x_{i0} среди составляющих решения системы уравнений (1.7) свидетельствует о нереализуемости плана производства, соответствующего выбранным предварительным оценкам производственных факторов. В этом случае следует уточнить систему оценок. Из числа базисных способов исключается r -й способ, отвечающий отрицательному значению x_{i0} .

Выбор r -го способа в качестве рентабельного послужил одной из причин нереализуемости плана производства. Уточнение предварительных оценок целесообразно провести таким образом, чтобы r -й способ стал нерентабельным, а остальные базисные способы по-прежнему оставались рентабельными. Увеличивая степень нерентабельности $(\Delta_r = \sum_{i=1}^m a_{ir} y_i - c_r)$ r -го способа, следует следить, чтобы система чисел y_1, y_2, \dots, y_m удовлетворяла свойствам (1.5) и (1.6) предварительных оценок. Полученная при этом предельная система предварительных оценок определит новый рентабельный способ производства, который войдет в число базисных способов вместо изъятого r -го. После этого необходимо снова проверить, может ли быть реализован план из базисных способов производства. Уточнение оценок производится до тех пор, пока векторы затрат рентабельных (с точки зрения очередной системы оценок) технологических способов не определяют реализуемый план производства.

Изложение экономической сущности метода сокращения невязок естественней всего производить на примере одной из модификаций метода, названной в гл. 6 методом двусторонних оценок.

Процесс решения задачи начинается с анализа некоторой системы предварительных оценок (плана сопряженной задачи) и плана производства (опорного плана исходной задачи).

Заданная система предварительных оценок определяет суммарную оценку расходуемых производственных факторов. Исходный производственный план определяет суммарную оценку выпускаемой продукции. Заданная система предварительных оценок и исходный план производства оптимальны, если суммарная оценка затрат совпадает с общей оценкой продукции. В противном случае определяется невязка плана — превышение расходов над доходами. Метод заключается в последовательном сокращении невязок.

Используя только базисные способы производства, отвечающие исходному опорному плану, и способы производства, рентабельные относительно заданной системы предварительных оценок, строим план производства, сводящий к минимуму невязку предыдущего плана. Этот план указывает направление изменения системы предварительных оценок. Если при измененных оценках производственных факторов суммарная оценка продукции совпадает с общей оценкой затрат, то полученный производственный план оптимален, а соответствующая система предварительных оценок оказывается планом цен. В противном случае решается очередная вспомогательная задача, в которой требуется минимизировать невязку ранее полученного плана на множестве способов производства, рентабельных с точки зрения соответствующей системы предварительных оценок или принадлежащих к базисным способам исследуемого плана. Таким образом, последовательное изменение предварительных оценок производственных факторов и системы базисных технологических способов постепенно сокращает невязку — превышение расходов над доходами — и приводит к выявлению способов производства и системы цен, наиболее целесообразных в условиях исследуемой задачи.

Аналогичное экономическое истолкование можно привести и для других модификаций метода сокращения невязок. Как видим, названия основных конечных методов линейного программирования полностью соответствуют их экономической сущности.

В методе последовательного улучшения плана и в различных его модификациях задается исходный план производства. Анализ плана позволяет установить его оптимальность или указать пути его улучшения. Для оптимального плана производства определяется система оценок производственных факторов — план цен.

В различных вариантах метода последовательного уточнения оценок исходным пунктом анализа является заданная система предварительных оценок производственных факторов. Попытки построения плана производства, соответствующего системе предварительных оценок, приводят к последовательному их уточнению. Системе оценок соответствует оптимальный план производства.

Наконец, в различных модификациях метода сокращения невязок процесс решения задачи начинают с плана производства и системы предварительных оценок, которые, вообще говоря, не согласованы между собой и приводят к невязке плана — превышению суммарной оценки затрат над суммарной оценкой продукции. Последовательное сокращение невязок приводит к оптимальному плану производства и системе оценок производственных факторов.

Таким образом, экономическое истолкование задачи линейного программирования позволяет характеризовать основные конечные методы как наиболее естественные пути составления оптимального плана работы предприятия и оценки его ресурсов.

1.5. В заключение параграфа рассмотрим некоторые формальные признаки методов линейного программирования, которые могут быть положены в основу их классификации.

Пусть задача записана в канонической форме. Требуется вычислить максимум линейной формы

$$L = CX \quad (1.8)$$

при условиях

$$AX = B, \quad (1.9)$$

$$X \geq 0. \quad (1.10)$$

Первая теорема двойственности позволяет заменить решение экстремальной задачи (1.8) — (1.10) вычислением n -мерного вектора X и m -мерного вектора Y , удовлетворяющих следующей системе линейных уравнений и линейных неравенств:

$$AX = B, \quad (1.11)$$

$$X \geq 0, \quad (1.12)$$

$$YA \geq C, \quad (1.13)$$

$$CX = BY. \quad (1.14)$$

Последнее равенство является следствием системы условий

$$(YA)_j = c_j \quad \text{при} \quad x_j \neq 0. \quad (1.15)$$

Четыре системы равенств и неравенств (1.11) — (1.14) определяют четыре группы методов линейного программирования. Переход от итерации к итерации в любом методе приводит к движению по векторам X и Y , для которых выполняется три из четырех систем условий, а четвертая система используется в качестве признака оптимальности пары (X, Y) .

Каждый шаг метода, не нарушая трех выполненных систем условий, обеспечивает монотонное в некотором смысле приближение к множеству точек (X, Y) , для которых выполняется четвертая система условий.

В методе *последовательного улучшения плана* переход от одной итерации метода к следующей соответствует движению по опорным планам X задачи (1.8) — (1.10). Следовательно, условия (1.11) и (1.12) удовлетворяются на каждом шаге метода. Каждому опорному плану X метод однозначно приводит в соответствие вектор Y , для которого выполняются условия (1.15), а следовательно, и равенство (1.14). Таким образом, метод улучшения плана связан с движением по точкам (X, Y) , для которых удовлетворяются условия (1.11), (1.12) и (1.14), а условие (1.13) не выполняется. Неравенства (1.13) или,

что то же самое, условия

$$\Delta_j = \sum_{i=1}^m a_{ij}y_i - c_j = \sum_{i=1}^m c_{s_i}x_{ij} - c_j \geq 0, \quad j = 1, 2, \dots, n,$$

определяют признак оптимальности метода.

Пусть на некотором шаге метода мы пришли в точку (X_0, Y_0) . Многогранное множество условий двойственной задачи задается системой неравенств (1.13). Рассмотрим в пространстве переменных двойственной задачи гиперплоскость

$$BY = CX_0 (=BY_0).$$

Нетрудно доказать (предоставим это читателю), что каждая итерация метода улучшения плана монотонно сокращает расстояние гиперплоскости $BY = CX_0$ до многогранного множества $YA \geq C$. Гиперплоскость, отвечающая оптимальному плану, является опорной для области определения линейной формы двойственной задачи.

Переход от итерации к итерации метода *последовательного уточнения оценок* соответствует движению по опорным планам двойственной задачи. Следовательно, условия (1.13) выполняются на каждом шаге метода. Каждый опорный план Y двойственной задачи определяет некоторый псевдоплан X исходной задачи, для которого удовлетворяются условия (1.11) и (1.14) (вследствие справедливости соотношений (1.15)). Таким образом, метод уточнения оценок связан с движением по точкам (X, Y) , для которых выполняются условия (1.11), (1.13) и (1.14). Условия (1.12) — неотрицательность составляющих псевдоплана — являются признаком оптимальности псевдоплана. При каждом шаге метода монотонно сокращается расстояние гиперплоскости $CX = BY_0 (=CX_0)$ от многогранного множества условий исходной задачи $AX = B, X \geq 0$. Гиперплоскость, отвечающая решению задачи (1.8) — (1.10), оказывается опорной для области определения линейной формы исходной задачи.

В методе *последовательного сокращения невязок* каждой итерации соответствует план Y сопряженной задачи и определяемый им квазиплан X исходной задачи. Для векторов X и Y удовлетворяются условия (1.12), (1.13) и (1.14). Нарушено только условие (1.11). Метод состоит

в движении по квазипланам X (и, следовательно, по планам Y сопряженной задачи), при котором монотонно сокращается невязка ε_0 квазиплана

$$\varepsilon_0 = \sum_{i=1}^m |(AX - B)_i|.$$

Схема вычислений упрощается, если ввести дополнительное предположение о неотрицательности невязок

$$\varepsilon_i = (B - AX)_i \geq 0, \quad i = 1, 2, \dots, m.$$

Условие (1.11) — равенство нулю невязок всех условий — признак оптимальности квазиплана.

Методы четвертой группы связаны с движением по точкам (X, Y) , отвечающим планам пары двойственных задач. Таким образом, на каждом шаге метода удовлетворяются условия (1.11), (1.12) и (1.13). Нарушено только условие (1.14). С каждым шагом метода разность $BY - CX$ монотонно сокращается.

Один из методов четвертой группы состоит в одновременном улучшении планов прямой и двойственной задач (см. п. 5.2 гл. 5). Векторы X и Y представляют собой в этом случае опорные планы пары двойственных задач. От итерации к итерации монотонно сокращается разность линейных форм обеих задач.

Другим примером методов четвертой группы является метод двусторонних оценок. Каждая итерация метода приводит к опорному плану X исходной задачи и плану Y сопряженной задачи. Точка (X, Y) удовлетворяет условиям (1.11), (1.12), (1.13), а разность $BY - CX$ от шага к шагу монотонно сокращается.

В каждом из перечисленных методов движение происходит по опорным планам, по псевдопланам или по квазипланам исходной задачи. Общее число опорных планов, псевдопланов и квазипланов задачи ограничено. Таким образом, для любого из рассмотренных методов общее количество точек (X, Y) , для которых удовлетворяются три из четырех систем условий (1.11) — (1.14), конечно. Кроме того, в невырожденных задачах (к этому случаю можно, как мы видели, искусственным приемом свести любую задачу линейного программирования) переход от

итерации к итерации связан с монотонным приближением к решению задачи. Оба указанных обстоятельства гарантируют достижение оптимального плана за конечное число шагов.

До сих пор в этом параграфе мы рассматривали задачи линейного программирования, записанные в канонической форме. Более общая запись задачи, в которой среди условий содержатся равенства и неравенства и не все переменные связаны условием неотрицательности, позволяет расщепить методы линейного программирования на большее число групп. Вместо двух систем условий (1.11) и (1.12), определяющих области изменения линейной формы прямой задачи, в более общем случае появляются три системы

$$A_1X = B_1; \quad A_2X \geq B_2; \quad x_j \geq 0; \quad j = j_1, j_2, \dots, j_t.$$

Увеличивается также число различных систем условий, определяющих область изменения переменных двойственной задачи (их становится также три). Неизменным остается только условие (1.14). Таким образом, решение задачи линейного программирования сводится к решению семи систем линейных уравнений и линейных неравенств. В новой классификации к методам одной группы относятся методы, соответствующие движению по векторам (X, Y) , для которых удовлетворяются шесть из семи систем условий. Седьмая система принимается в качестве признака оптимальности векторов (X, Y) . Каждый шаг метода обеспечивает монотонное в некотором смысле приближение к оптимальным планам пары двойственных задач.

Подчеркнем, что приведенные в настоящем пункте характеристики методов могут быть использованы также для классификации итеративных методов линейного программирования. Однако в этом случае набор точек (X, Y) , по которым осуществляется движение, бесконечен. Поэтому итеративные методы дают лишь приближенное (с любой заранее заданной степенью точности) решение задачи.

Формальная классификация, принятая здесь для методов линейного программирования, обеспечивает также

единый подход ко всем существующим методам решения нелинейных условных экстремальных задач и может быть поэтому использована в качестве основы для систематизации методов математического программирования.

§ 2. МОДИФИКАЦИЯ КОНЕЧНЫХ МЕТОДОВ

2.1. Рассмотрим общую задачу линейного программирования, заданную в канонической форме.

Метод последовательного улучшения плана состоит в движении по соседним опорным планам задачи, базисы которых различаются единственным вектором. Вектор, подлежащий включению в базис, выбирается из числа векторов условий, имеющих отрицательные относительные оценки. Обычно этим свойством обладает ряд векторов условий. Каждый такой вектор является подходящим для введения в базис. Однако в соответствии с рекомендациями метода улучшения плана в новый базис включается только один подходящий вектор. Возникает вопрос, нельзя ли так модифицировать метод улучшения плана, чтобы можно было вводить в базис сразу несколько подходящих векторов условий. Аналогичный вопрос встает и для метода последовательного уточнения оценок. При использовании метода уточнения оценок процесс решения задачи заключается в движении по ее соседним псевдопланам. Базисы двух соседних псевдопланов различаются только одним вектором условий. Для перехода к новому псевдоплану вычисляются коэффициенты разложения вектора ограничений по векторам базиса данного псевдоплана. Векторы базиса, отвечающие отрицательным коэффициентам, являются подходящими для исключения из базиса. Хотя в большинстве случаев имеется несколько подходящих для исключения векторов, из базиса удаляется только один из них. Поэтому совершенно естественным является вопрос о том, можно ли исключить из базиса псевдоплана сразу несколько подходящих векторов.

Настоящий параграф посвящен рассмотрению поставленных вопросов. Оказывается, каждый из них разрешается в положительном смысле, причем для задач больших размеров реализация соображений, указанных ниже, приводит к весьма ощутимым вычислительным преимуществам.

Основная часть параграфа относится к методу улучшения плана, модификация метода уточнения оценок излагается более кратко.

2.2. Все рассуждения будут проводиться применительно к задаче линейного программирования, состоящей в максимизации линейной формы

$$L(X) = \sum_{j=1}^n c_j x_j \quad (2.1)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (2.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (2.3)$$

Пусть $X_0 = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ — опорный план задачи (2.1) — (2.3) с базисом $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Здесь, как обычно, $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$ — j -й вектор условий задачи (2.1) — (2.3).

Перепишем систему уравнений (2.2) в эквивалентной форме, выразив базисные переменные данного плана относительно остальных неизвестных системы

$$x_{s_i} = x_{i0} - \sum_{j \notin I_{X_0}} x_{ij} x_j, \quad i = 1, 2, \dots, m. \quad (2.4)$$

Здесь величины $x_{1j}, x_{2j}, \dots, x_{mj}$ являются соответствующими коэффициентами разложения вектора A_j по векторам $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ базиса плана X_0 ($j = 0, 1, 2, \dots, n$; $A_0 = E_j$). Через I_{X_0} обозначено множество индексов s_1, s_2, \dots, s_m , отвечающих векторам базиса плана X_0 .

Если заменить в линейной форме (2.1) переменные x_{s_i} , $i = 1, 2, \dots, m$, их выражениями из (2.4), то получим

$$L(X) = L(X_0) - \sum_{j \notin I_{X_0}} \Delta_j x_j, \quad (2.5)$$

где

$$\Delta_j = \sum_{i=1}^m x_{ij} c_{s_i} - c_j, \quad j = 1, 2, \dots, n.$$

В случае $\Delta_j \geq 0$ для $j \notin I_{X_0}$ план X_0 оказывается оптимальным. Допустим теперь, что среди параметров Δ_j ,

имеются отрицательные. В соответствии с методом улучшения плана выбирается один из таких параметров, скажем $\Delta_k < 0$. После этого внебазисная переменная x_k начинает увеличиваться. Остальные внебазисные переменные сохраняют свои нулевые значения. Величины базисных переменных определяются формулой (2.4).

Если обозначить текущее значение k -й переменной через θ , то для выполнения условий (2.3) необходимо и достаточно, чтобы

$$\left. \begin{aligned} x_{i0} - \theta x_{ik} &\geq 0 \quad \text{для } i = 1, 2, \dots, m, \\ \theta &\geq 0. \end{aligned} \right\} \quad (2.6)$$

При соблюдении требований (2.6) вектор

$$X(\theta) = (x_1(\theta), x_2(\theta), \dots, x_n(\theta)),$$

где

$$x_j(\theta) = \begin{cases} x_{i0} - \theta x_{ik}, & j = s_i \quad (i = 1, 2, \dots, m), \\ \theta, & j = k, \\ x_j^{(0)} = 0 & \text{в остальных случаях,} \end{cases}$$

оказывается планом задачи (2.1) — (2.3).

Переход от плана X_0 к плану $X(\theta)$ сопровождается возрастанием линейной формы (2.1) на величину

$$L(X(\theta)) - L(X_0) = -\Delta_k \theta. \quad (2.7)$$

Естественно подобрать такое значение θ_0 переменной θ , на котором линейная функция (2.7) достигает максимума при условиях (2.6). Значение θ_0 является, таким образом, решением задачи линейного программирования (2.6), (2.7) с одной переменной (θ). Допустим теперь, что имеется ряд отрицательных величин Δ_j . Попытаемся ввести сразу несколько векторов условий $A_j (j \notin I_{X_0}, \Delta_j < 0)$ в базис плана X_0 .

Обозначим через E множество индексов векторов A_μ , которые выбраны для одновременного включения в базис. Пусть $\theta_\mu (\mu \in E)$ — текущее значение переменной, отвечающей вводимому вектору A_μ . Из соотношений (2.4) получаем, что при фиксированных значениях $\theta_\mu (\mu \in E)$ величины базисных (в смысле X_0) переменных x_{s_i} оказываются

равными

$$x_{i0} - \sum_{\mu \in E} x_{i\mu} \theta_{\mu}.$$

Таким образом, параметры θ_{μ} , $\mu \in E$, определяют преобразование плана X_0 в вектор $X(\theta) = (x_1(\theta), \dots, x_n(\theta))$, определяемый формулой

$$x_j(\theta) = \begin{cases} x_{i0} - \sum_{\mu \in E} x_{i\mu} \theta_{\mu}, & j = s_i \quad (i = 1, 2, \dots, m), \\ \theta_j, & j \in E, \\ 0 & \text{в остальных случаях.} \end{cases} \quad (2.8)$$

Здесь под θ понимается набор параметров θ_{μ} при $\mu \in E$.

По построению вектор $X(\theta)$ удовлетворяет равенствам (2.2) при любых значениях θ_{μ} , $\mu \in E$. Поэтому соблюдение системы неравенств

$$x_{i0} - \sum_{\mu \in E} x_{i\mu} \theta_{\mu} \geq 0, \quad i = 1, 2, \dots, m, \quad (2.9)$$

и

$$\theta_{\mu} \geq 0, \quad \mu \in E, \quad (2.10)$$

является необходимым и достаточным условием для того, чтобы вектор $X(\theta)$ был планом задачи (2.1) — (2.3).

Из равенства (2.5) получаем

$$L(X(\theta)) - L(X_0) = - \sum_{\mu \in E} \Delta_{\mu} \theta_{\mu}. \quad (2.11)$$

Параметры θ_{μ} , $\mu \in E$, естественно выбрать такими, чтобы при переходе к новому плану $X(\theta)$ линейная форма (2.1) увеличилась на возможно большую величину. Следовательно, искомые значения параметров θ_{μ} , $\mu \in E$, определяются путем максимизации линейной формы (2.11), переменные которой связаны ограничениями (2.9) и (2.10).

Назовем задачу (2.9) — (2.11) *вспомогательной задачей*, отвечающей плану X_0 и множеству E . Обозначим через t число переменных вспомогательной задачи, равное числу элементов множества E . Матрица $A_{X_0, E}$ условий вспомогательной задачи состоит из элементов $x_{i\mu}$ и имеет размеры $m \times t$.

2.3. Сформулированная задача (2.9)—(2.11) записана не в канонической форме. Поэтому для ее решения может быть использована модификация метода улучшения плана, изложенная в § 1 гл. 7. В соответствии с общим определением каждому опорному плану задачи (2.9)—(2.11) отвечает невырожденная квадратная подматрица матрицы A_{X_0} , E — базис опорного плана. Порядок базиса $\bar{\tau}$ изменяется в пределах от 0 до $\min(t, m)$. Число переменных t вспомогательной задачи (2.9)—(2.11) обычно выбирается меньшим m . Поэтому $0 \leq \bar{\tau} \leq t$.

Отметим, что случай $\bar{\tau} = 0$ (множество элементов базиса пусто) в данной задаче реализуем. Этому случаю отвечает опорный план: $\theta_\mu = 0$ для $\mu \in E$.

Приняв опорный план $\theta_\mu = 0$ для $\mu \in E$ за исходный, применим для решения задачи (2.9)—(2.11) метод последовательного улучшения плана. Поскольку $\Delta_\mu < 0$ при $\mu \in E$, первая итерация метода не может завершиться случаем 1°. Следовательно, процесс решения вспомогательной задачи закончится либо установлением ее неразрешимости, либо получением опорного оптимального плана с базисом порядка τ , где $1 \leq \tau \leq t$.

Первая возможность указывает на неразрешимость исходной задачи (2.1)—(2.3). Рассмотрим вторую возможность.

Пусть базис полученного опорного решения $\bar{\theta}_\mu$, $\mu \in E$, вспомогательной задачи состоит из элементов, расположенных на пересечении строк и столбцов матрицы A_{X_0}, E с номерами r_1, r_2, \dots, r_τ и k_1, k_2, \dots, k_τ соответственно. Обозначим через $X' = (x'_1, x'_2, \dots, x'_n)$ план задачи (2.1)—(2.3), который определяется по формуле (2.8), где параметры θ_μ заменены на $\bar{\theta}_\mu$ ($\mu \in E$).

Очевидно,

$$x'_{s_i} = x_{i0} - \sum_{\mu \in E} x_{l_\mu} \bar{\theta}_\mu = 0 \text{ при } i = r_1, r_2, \dots, r_\tau,$$

$$x'_j = 0, \text{ если } j \notin I_{X_0} \text{ и } j \neq k_1, k_2, \dots, k_\tau.$$

Пусть $I_{X'}$ — множество индексов векторов условий, образуемое из I_{X_0} заменой индексов s_i для $i = r_1, r_2, \dots, r_\tau$ на k_1, k_2, \dots, k_τ .

Для доказательства опорности плана X' необходимо убедиться в линейной независимости векторов A_j , $j \in I_{X'}$.

Положим $X_j = A_{X_0}^{-1} A_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T$, где A_{X_0} — матрица, составленная из векторов базиса плана X_0 . Образует определитель $D_{X'}$ из компонент векторов X_j , $j \in I_{X'}$. Разлагая определитель $D_{X'}$ по столбцам

$$X_{s_i} = (0, 0, \underbrace{\dots, 0}_i, 1, 0, \dots, 0)^T$$

для $1 \leq i \leq m$, $i \neq r_\lambda$, $1 \leq \lambda \leq \tau$, имеем

$$D_{X'} = \pm D_0,$$

где D_0 — определитель базиса решения вспомогательной задачи (2.9) — (2.11). Но $D_0 \neq 0$. Следовательно, $D_{X'} \neq 0$ и система векторов X_j , $j \in I_{X'}$, линейно независима.

Учитывая далее, что $A_j = A_{X_0} X_j$ и $|A_{X_0}| \neq 0$, приходим к выводу о линейной независимости системы A_j для $j \in I_{X'}$. Итак, опорность плана X' доказана.

В результате решения вспомогательной задачи получаем:

а) индексы $j \in E$ векторов A_j , которые целесообразно включить в базис ($j = k_\lambda$, $\lambda = 1, 2, \dots, \tau$);

б) индексы s_i векторов A_{s_i} , подлежащих исключению из базиса ($i = r_\lambda$, $\lambda = 1, 2, \dots, \tau$).

Отметим, что число τ векторов, вводимых в базис, может оказаться меньшим числа t элементов множества E . Однако, как мы видели, $\tau \geq 1$. Поэтому хотя бы одна позиция базиса будет заведомо обновлена. В большинстве случаев число обновляемых позиций базиса оказывается большим единицы. Совокупность операций, необходимых для перехода от плана X_0 к плану X' , назовем *большой итерацией*. Отдельный шаг метода улучшения плана, применяемого для решения вспомогательной задачи, будем называть *малой итерацией*. Для начала очередной большой итерации, связанной с опорным планом X' , необходимо знать либо коэффициенты разложения векторов условий по базису плана X' (первый алгоритм), либо матрицу, обратную для базиса плана X' (второй алгоритм).

2.4. Остановимся на втором алгоритме. Выведем рекуррентные соотношения, которые связывают элементы матриц $A_{X_0}^{-1} = \|e_{ij}\|_m$ и $A_{X'}^{-1} = \|e'_{ij}\|_m$.

Матрицы A_{X_0} и $A_{X'}$ — матрицы базисов планов X_0 и X' — различаются τ векторами. Для упрощения обозначений допустим, что в матрице A_{X_0} заменяются первые τ позиций, т. е. $r_1 = 1, \dots, r_\tau = \tau$. Этого всегда можно добиться за счет соответствующей перестановки столбцов матрицы A_{X_0} и строк матрицы $A_{X_0}^{-1}$. Итак,

$$A_{X'} = (A_{k_1}, A_{k_2}, \dots, A_{k_\tau}, A_{s_{\tau+1}}, \dots, A_{s_m}).$$

Умножая $A_{X'}$ слева на $A_{X_0}^{-1}$, имеем

$$A_{X_0}^{-1} A_{X'} = (X_{k_1}, X_{k_2}, \dots, X_{k_\tau}, e_{\tau+1}, \dots, e_m), \quad (2.12)$$

где

$$X_j = A_{X_0}^{-1} A_j, \quad e_i = (\underbrace{0, \dots, 0}_i, 1, 0, \dots, 0)^T.$$

Матрица $(X_{k_1}, X_{k_2}, \dots, X_{k_\tau})$ составлена из коэффициентов разложения вводимых в базис векторов $A_{k_1}, A_{k_2}, \dots, A_{k_\tau}$ по векторам базиса плана X_0 .

Пусть \overline{B}_τ — квадратная матрица порядка τ , образованная первыми τ строками матрицы $(X_{k_1}, X_{k_2}, \dots, X_{k_\tau})$ (\overline{B}_τ состоит из коэффициентов разложения вводимых векторов, которые соответствуют векторам, исключаемым из базиса). Обозначим через $\overline{\overline{B}}_{m-\tau, \tau}$ матрицу размеров $(m-\tau) \times \tau$, составленную из остальных $(m-\tau)$ строк матрицы $(X_{k_1}, X_{k_2}, \dots, X_{k_\tau})$. В этих обозначениях равенство (2.12) может быть переписано в виде

$$A_{X_0}^{-1} A_{X'} = \begin{bmatrix} \overline{B}_\tau & 0_{\tau, m-\tau} \\ \overline{\overline{B}}_{m-\tau, \tau} & E_{m-\tau} \end{bmatrix}, \quad (2.13)$$

где $0_{\tau, m-\tau}$ — нулевая матрица размеров $\tau \times (m-\tau)$; $E_{m-\tau}$ — единичная матрица порядка $m-\tau$.

Путем непосредственного перемножения матриц нетрудно проверить справедливость равенства

$$D = \begin{bmatrix} \overline{B}_\tau & 0_{\tau, m-\tau} \\ \overline{\overline{B}}_{m-\tau, \tau} & E_{m-\tau} \end{bmatrix}^{-1} = \begin{bmatrix} \overline{B}_\tau^{-1} & 0_{\tau, m-\tau} \\ -\overline{\overline{B}}_{m-\tau, \tau} \overline{B}_\tau^{-1} & E_{m-\tau} \end{bmatrix}. \quad (2.14)$$

Используя равенства (2.13) и (2.14), получаем

$$DA_{X_0}^{-1}A_{X'} = E_m.$$

Следовательно,

$$A_{X'}^{-1} = DA_{X_0}^{-1}. \quad (2.15)$$

Соотношение (2.15) устанавливает искомую связь между матрицами $A_{X'}^{-1}$ и $A_{X_0}^{-1}$.

Разделим элементы матрицы $A_{X_0}^{-1} = \|e_{ij}\|_m$ на две группы: первая группа охватывает первые τ строк матрицы $A_{X_0}^{-1}$, вторая составлена из элементов оставшихся $m - \tau$ строк. Полученные группы элементов представляют собой матрицы $\|e_{ij}\|_{\tau, m}$, $\|e_{i+\tau, j}\|_{m-\tau, m}$ размеров $\tau \times m$ и $(m - \tau) \times m$ соответственно.

Аналогично разобьем матрицу $A_{X'}^{-1} = \|e'_{ij}\|_m$ на матрицы $\|e'_{ij}\|_{\tau, m}$ и $\|e'_{i+\tau, j}\|_{m-\tau, m}$.

Учет специальной структуры матрицы D , определяемой соотношением (2.14), позволяет конкретизировать формулу (2.15):

$$\|e'_{i+\tau, j}\|_{m-\tau, m} = \|e_{i+\tau, j}\|_{m-\tau, m} - \overline{B}_{m-\tau, \tau} \overline{B}_{\tau}^{-1} \|e_{ij}\|_{\tau, m}, \quad (2.16)$$

$$\|e'_{ij}\|_{\tau, m} = \overline{B}_{\tau}^{-1} \|e_{ij}\|_{\tau, m}. \quad (2.17)$$

Полученные формулы являются естественным обобщением рекуррентных соотношений (5.21) гл. 4, которые широко использовались при алгоритмизации методов линейного программирования. Эти соотношения — частный случай формул (2.16), (2.17) при $\tau = 1$.

Приведем краткое описание последовательности действий, связанных с одной большой итерацией при использовании второго алгоритма. Перед началом итерации известны опорный план X_0 с базисом $(A_{s_1}, A_{s_2}, \dots, A_{s_m}) = A_{X_0}$ и обратная матрица $A_{X_0}^{-1} = \|e_{ij}\|_m$.

Вычисляются величины

$$\lambda_i = \sum_{\alpha=1}^m c_{s_\alpha} e_{\alpha i}, \quad i = 1, 2, \dots, m.$$

Затем определяются параметры

$$\Delta_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j \quad \text{для } j \notin I_{X_0}.$$

В случае неотрицательности всех параметров Δ_j делается вывод об оптимальности плана X_0 . Если же среди Δ_j имеются отрицательные числа, то приступают к построению нового опорного плана X' . Для этого выбирается несколько (t) векторов условий, которые целесообразно ввести в базис. (Множество индексов таких векторов обозначено через E .) Целесообразность ввода вектора A_j в базис обычно определяется величиной Δ_j — оценки A_j относительно старого базиса. Однако при выборе системы вводимых векторов могут использоваться и другие соображения. Выбор числа t элементов множества E во многом зависит от характера применяемых вычислительных средств, обычно $t < m$. (К этому вопросу мы вернемся в следующем пункте.)

Множество E определяет вспомогательную задачу (2.9) — (2.11) с t неотрицательными переменными и m условиями-неравенствами. Параметры x_{ij} вспомогательной задачи вычисляются с помощью матрицы $\|e_{ij}\|_m$:

$$x_{ij} = \sum_{\alpha=1}^m a_{\alpha j} e_{i\alpha} \quad (j \in E).$$

Для решения задачи (2.9) — (2.11) применяется модификация метода улучшения плана, описанная в § 1 гл. 7. В исходном плане вспомогательной задачи все переменные равны нулю, базис этого плана не содержит ни одного элемента. На каждом шаге решения вспомогательной задачи вычисляется (по рекуррентным формулам) обратная матрица по отношению к базису соответствующего плана.

Процесс решения завершается либо обнаружением неразрешимости вспомогательной задачи, либо определением ее оптимального плана с базисом \bar{B}_τ . В первом случае делается вывод о неразрешимости задачи (2.1) — (2.3). Во втором — осуществляется переход к новому опорному плану исследуемой задачи.

Пусть базис \bar{B}_τ оптимального плана $\bar{\theta}_\mu$, $\mu \in E$, задачи (2.9) — (2.11) состоит из элементов матрицы условий вспомогательной задачи, которые расположены на пересечении строк и столбцов с номерами r_1, r_2, \dots, r_τ и k_1, k_2, \dots, k_τ соответственно. В таком случае базис A_X

нового опорного плана X' задачи (2.1) — (2.3) образуется из A_{X_0} путем обновления позиций r_1, r_2, \dots, r_τ за счет векторов $A_{k_1}, A_{k_2}, \dots, A_{k_\tau}$; компоненты плана X' вычисляются по формулам (2.8), где θ_μ заменены на $\bar{\theta}_\mu$.

При решении вспомогательной задачи вместе с ее оптимальным планом вычисляется матрица \bar{B}_τ^{-1} . Поэтому для определения матрицы $A_{X'}^{-1} = \|e'_{ij}\|_m$, обратной для базиса плана X' , можно воспользоваться рекуррентными соотношениями (2.16), (2.17). В результате получаем план X' и матрицу $A_{X'}^{-1}$ — необходимые данные для проведения следующей большой итерации.

2.5. Сделаем несколько замечаний, которые позволят читателю более отчетливо представить смысл и достоинства описанной здесь модификации метода улучшения плана. Поясним геометрическую сущность большой итерации. Ограничимся первой геометрической интерпретацией. Опорный план X_0 соответствует вершине многогранного множества условий задачи (2.1) — (2.3) (множества M). После выяснения неоптимальности плана X_0 формируется множество E индексов векторов условий, выбранных для включения в новый базис. Множество E определяет вспомогательную задачу данной итерации. Вершина X_0 является пересечением гиперплоскостей с уравнениями (2.2) и

$$x_j = 0 \text{ при } j \notin I_{X_0}. \quad (2.18)$$

Выделив множество E , мы заменяем ограничения (2.18) для $j \in E$ условиями $x_j \geq 0$. Следовательно, многогранное множество M_E условий вспомогательной задачи является гранью множества M , проходящей через вершину X_0 . Вспомогательная задача связана с максимизацией линейной формы основной задачи на множестве M_E . Размерность ρ_E множества M_E не превосходит t . В невырожденном случае $\rho_E = t$. Точка X_0 — вершина множества M_E . Процесс решения вспомогательной задачи состоит в движении по соседним вершинам множества M_E , начиная с вершины X_0 . Это движение осуществляется по правилам метода улучшения плана и поэтому (в невырожденном

случае) сопровождается монотонным увеличением линейной формы задачи (2.1)—(2.3).

Операции, связанные с переходом от одной вершины множества M_E к соседней, составляют малую итерацию. Каждая вершина M_E является вершиной множества M . Поэтому, получив оптимальный опорный план вспомогательной задачи, мы тем самым придем к новому плану X' основной задачи — результату данной большой итерации.

При $t=1$ и невырожденности плана X_0 множество M_E оказывается ребром множества M . Ограниченное множество M_E имеет две вершины. Поэтому решение вспомогательной задачи требует только одной итерации. Этот случай имеет место при использовании обычной формы метода улучшения плана.

Вспомогательная задача образуется из основной задачи, если в последней некоторые из переменных полагаются равными нулю. Выбор нулевых переменных для каждой вспомогательной задачи может осуществляться различными способами. При использовании метода улучшения плана естественно руководствоваться величинами оценок Δ_j векторов условий A_j относительно данного базиса, полагая равными нулю те x_j , для которых $\Delta_j > \Delta$, где Δ — некоторое число. В методе сокращения невязок и в его модификациях также строится последовательность вспомогательных задач. Однако здесь для выбора $x_j = 0$ используется текущий план двойственной задачи.

Приведенная здесь модификация метода улучшения плана позволяет расширить ограничения на размеры задачи, которые определяются емкостью оперативной памяти используемой универсальной вычислительной машины. Матрица условий задачи больших размеров записывается во внешнюю память вычислительной машины. В процессе проведения большой итерации в оперативной памяти машины целесообразно хранить только условия вспомогательной задачи, размеры которой $m \times t$. Это позволит отнести все взаимодействия с внешней памятью в конец большой итерации. Значительная часть времени, затрачиваемого машиной на решение задачи, связана с обращением к внешним запоминающим устройствам. Поэтому сокращение числа таких обращений резко уменьшает время решения задачи.

При формировании вспомогательной задачи имеется возможность варьировать значением параметра t — числом переменных вспомогательной задачи. Емкость оперативной памяти ограничивает величину t сверху. В начале решения основной задачи, когда имеющийся план еще далек от оптимального, приращение линейной формы за одну большую итерацию оказывается тем значительнее, чем больше число t . Поэтому, увеличивая t , мы сокращаем количество больших итераций и, следовательно, уменьшаем число обращений к внешней памяти.

Естественно, что при этом возрастает количество малых итераций, необходимых для решения вспомогательной задачи. Однако увеличение размеров вспомогательной задачи мало сказывается на времени, потребном для решения всей задачи, так как малые итерации не требуют обращения к внешней памяти. Итак, в начале процесса решения задачи следует выбирать значение параметра t возможно большим. В конце решения основной задачи параметру t целесообразно придавать несколько меньшее значение, поскольку приращение линейной формы теперь уже мало зависит от t .

При решении задач больших размеров конечными методами линейного программирования число итераций может быть достаточно велико. Поэтому для получения искомого решения с определенной степенью точности следует:

а) либо проводить все вычисления с достаточным запасом знаков,

б) либо через каждые несколько итераций вычислять текущие параметры непосредственно (без использования рекуррентных формул).

Использование первого пути ограничивается емкостью оперативной памяти. Реализация второй рекомендации может существенно увеличить время, необходимое для решения задачи. Применение предложенной здесь модификации метода улучшения плана снижает влияние ошибок округления. Это объясняется тем, что в течение одной большой итерации решается задача сравнительно малых размеров. При переходе к следующей большой итерации параметры задачи иногда целесообразно пересчитывать путем непосредственного вычисления соответствующей обратной матрицы.

2.6. Рассмотрим теперь возможность одновременного вывода нескольких векторов из базиса псевдоплана при использовании метода уточнения оценок. Описание соответствующей модификации метода уточнения оценок проводится кратко. Тем не менее читатель, познакомившись с предшествующими пунктами этого параграфа, сможет восстановить недостающие детали самостоятельно.

Допустим, что нам известен псевдоплан $X_0 = (x_1^{(0)}, \dots, x_n^{(0)})$ задачи (2.1) — (2.3) с базисом $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Псевдоплан X_0 отвечает некоторому опорному плану $Y_0 = (y_1^{(0)}, y_2^{(0)}, \dots, y_m^{(0)})$ задачи, двойственной по отношению к задаче (2.1) — (2.3).

Пусть

$$A_j = \sum_{i=1}^m x_{ij} A_{s_i}, \quad j = 0, 1, \dots, n,$$

$$\Delta_j = \sum_{i=1}^m a_{ij} y_i^{(0)} - c_j = \sum_{i=1}^m c_{s_i} x_{ij} - c_j, \quad j = 1, 2, \dots, n.$$

Таким образом,

$$x_{s_i}^{(0)} = x_{i0}, \quad i = 1, 2, \dots, m, \quad x_j^{(0)} = 0 \quad \text{при} \quad j \neq s_i.$$

Предположим, что среди базисных компонент x_{i0} псевдоплана X_0 имеются отрицательные. Это указывает на необходимость изменения X_0 путем уточнения системы предварительных оценок Y_0 . Выберем несколько векторов, входящих в базис псевдоплана X_0 и имеющих отрицательные значения базисных компонент.

Пусть эти векторы расположены в позициях базиса с номерами r_1, r_2, \dots, r_t ($t < m$). Для того чтобы выяснить, какие из выбранных векторов целесообразно исключить из базиса и какими векторами следует заполнить освободившиеся позиции, необходимо решить следующую вспомогательную задачу: максимизировать линейную форму

$$- \sum_{j=1}^n \Delta_j x_j \quad (2.19)$$

при соблюдении условий

$$\sum_{j=1}^n x_{ij} x_j = x_{i0}, \quad i = r_1, r_2, \dots, r_t, \quad (2.20)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (2.21)$$

Пусть компоненты вектора $\bar{X}_0 = (\bar{x}_1^{(0)}, \bar{x}_2^{(0)}, \dots, \bar{x}_n^{(0)})$ определяются равенствами

$$\bar{x}_j^{(0)} = \begin{cases} x_j^{(0)}, & \text{если } j = s_\lambda \quad \text{при } \lambda = r_1, r_2, \dots, r_t, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Как нетрудно видеть, \bar{X}_0 —псевдоплан задачи (2.19)—(2.21) с базисом A_{s_λ} , $\lambda = r_1, r_2, \dots, r_t$. Поэтому для решения задачи (2.19)—(2.21) целесообразно использовать метод уточнения оценок, отправляясь от имеющегося псевдоплана \bar{X}_0 . Решив задачу (2.19)—(2.21), получим ее оптимальный план с базисом, составленным из векторов $A_{k_1}, A_{k_2}, \dots, A_{k_t}$. Базис нового псевдоплана задачи (2.1)—(2.3) образуется из предшествующего базиса путем обновления позиций r_1, r_2, \dots, r_t с помощью векторов $A_{k_1}, A_{k_2}, \dots, A_{k_t}$. Отметим, что некоторые из векторов A_{k_i} могут совпадать с векторами A_{s_λ} , $\lambda = r_1, \dots, r_t$; соответствующие позиции базиса не обновляются.

При использовании описанной модификации метода уточнения оценок можно придерживаться как первого, так и второго алгоритма. Предлагаемая модификация имеет те же вычислительные достоинства, что и соответствующее видоизменение метода улучшения плана (см. п. 2.5).

ИТЕРАТИВНЫЕ МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

До сих пор мы рассматривали конечные методы, позволяющие получить точное решение задачи линейного программирования за конечное число шагов. В последние годы много внимания уделяется итеративным методам решения условных экстремальных задач.

Итеративные методы представляют собой последовательности однообразных по процедуре выполнения итераций, приводящих в пределе к оптимальному плану задачи. Итеративные методы позволяют получить лишь приближенное решение задачи. При этом качество решения существенно зависит от числа проведенных итераций. Некоторые итеративные методы линейного программирования используют идеи решения задач, возникшие в других математических дисциплинах.

В частности, вычислительные приемы линейного программирования обогатились методами решения игр. Связи теории игр и линейного программирования посвящен § 1. В § 2 описаны два игровых итеративных метода и их модификации. Учитывая утверждения § 1, можно использовать эти методы для решения задач линейного программирования.

В § 3 изложены итеративные методы, разработанные непосредственно для решения условных экстремальных задач. В частности, это методы, основанные на сведениях при помощи «штрафных функций» условных экстремальных задач к безусловным и использовании градиентных приемов безусловной оптимизации функций многих переменных. Кроме того, в § 3 сравниваются конечные и итеративные методы и приводятся соображения в пользу комбинированных методов, позволяющих использовать достоинства тех и других методов линейного программирования.

§ 1. ТЕОРИЯ ИГР И ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

1.1. Теория игр, как и линейное программирование, вызвана к жизни практическими потребностями в моделях и методах экономического и военного планирования. Вначале обе эти дисциплины развивались независимо одна от другой по своим специфическим направлениям. В 1951 г. Данцигом [14] была обнаружена замечательная связь между линейным программированием и теорией игр. Переплетение и взаимное проникновение этих математических дисциплин оказались полезными для каждой из них.

Теория игр занимается изучением конфликтных ситуаций. Интересы игроков (участников конфликтных ситуаций) не совпадают, а иногда и прямо противоположны. При этом каждый из игроков может оказывать некоторое влияние на ход событий, но не способен полностью им управлять.

Мы здесь будем рассматривать простейшие так называемые прямоугольные (или матричные) игры двух лиц с нулевой суммой. Такие игры задаются платежной матрицей (или матрицей выигрышей) $\|a_{ij}\|_{m,n}$.

Первый игрок имеет возможность сделать m выборов (m чистых стратегий), второй — n выборов (n чистых стратегий). Если первый игрок выберет i -ю чистую стратегию, а второй — j -ю, то выигрыш первого (проигрыш второго) равен a_{ij} . Сумма выигрышей обоих игроков равна нулю.

Задача теории игр заключается в выработке принципов, определяющих поведение игроков в каждой конкретной ситуации. Решение игры — это выбор линий поведения игроков, обеспечивающих состояние равновесия, т. е. состояние, к которому стремился бы каждый разумный игрок, сознавая, что отступление от этой линии может только уменьшить его выигрыш.

В большом числе конфликтных ситуаций невозможно указать такие чистые стратегии игроков, которые обеспечивали бы ситуацию равновесия независимо от поведения противников. Однако теория позволяет выработать такую линию поведения, придерживаясь которой в каждой партии, каждый игрок может обеспечить ситуацию равновесия в среднем (для многих партий) независимо от поведения противника. Если один из противников не

будет в процессе последовательного повторения партий придерживаться правил оптимального выбора стратегий, средний выигрыш другого противника может увеличиться. Но каждый из игроков всегда может придерживаться таких правил выбора стратегий, которые не позволят противнику превзойти некоторый средний выигрыш в большом числе партий.

В играх с неполной информацией обычно теряет смысл разговор о какой бы то ни было фиксированной наиболее разумной линии поведения игроков в каждой партии. Как правило, невозможно обеспечить ситуацию равновесия независимо от поведения противника и в том случае, если в последовательных партиях применять определенную последовательность чистых стратегий. Противник после некоторого числа партий изучит закономерности нашего поведения и воспользуется этим для выбора своей стратегии. Скрытность нашего выбора линии поведения будет наилучшим образом обеспечена в том случае, если для выбора стратегии в каждой партии используется некоторый случайный механизм.

Приведенные интуитивные соображения сводят вопрос о выборе оптимальной линии поведения в игре к выбору статистических характеристик случайных механизмов, которые следует применять для выбора стратегий в каждой партии. Другими словами, выбор оптимальной стратегии сводится к выбору частоты, с которой следует использовать каждую чистую стратегию в игре.

Вектор $U = (u_1, u_2, \dots, u_m)$, каждая компонента которого указывает относительную частоту (вероятность), с которой соответствующая чистая стратегия используется в игре, называется *смешанной стратегией* первого игрока. Набор чисел $W = (w_1, w_2, \dots, w_n)$ — смешанная стратегия второго игрока. Ясно, что

$$u_i \geq 0, \quad i = 1, 2, \dots, m, \quad w_j \geq 0, \quad j = 1, 2, \dots, n,$$

$$\sum_{i=1}^m u_i = 1, \quad \sum_{j=1}^n w_j = 1.$$

Чистая стратегия может быть определена как смешанная стратегия, в которой все составляющие, кроме одной, равны нулю.

В дальнейшем будем обозначать чистые стратегии обоих противников в виде единичных векторов

$$e_i = (\underbrace{0, \dots, 0}_i, \underbrace{1, 0, \dots, 0}_m) \text{ и } e_j = (\underbrace{0, \dots, 0}_j, \underbrace{1, 0, \dots, 0}_n)$$

соответственно.

Оптимальная стратегия игрока — это стратегия, обеспечивающая ему максимально возможный гарантированный средний выигрыш. При этом предполагается, что соответствующая развернутая игра ведется без обмана и подглядываний. Всякое изменение информации приводит к новой игре, для которой оптимальная линия поведения будет иной.

Если первый игрок с вероятностью u_i выбирает i -ю чистую стратегию, а второй с вероятностью w_j — свою j -ю чистую стратегию, то средний платеж первому игроку равен

$$M(U, W) = \sum_{j=1}^n \sum_{i=1}^m a_{ij} u_i w_j = UAW,$$

где $A = \|a_{ij}\|_{m,n}$ — матрица выигрышей (платежная матрица игры). Соответственно выигрыш второго игрока равен $-M$. Функция $M(U, W)$ называется *платежной функцией*.

Будем говорить, что *игра имеет решение в смешанных стратегиях*, если существуют такие оптимальные стратегии U^* и W^* и число v , что при любых смешанных стратегиях U и W выполняются соотношения

$$M(U, W^*) \leq v \leq M(U^*, W). \quad (1.1)$$

Полагая $U = U^*$ и $W = W^*$, получаем

$$v = M(U^*, W^*). \quad (1.2)$$

Число v называется *ценой игры*.

Процесс вычисления оптимальных стратегий U^* и W^* и цены игры v называется *процессом решения игры* или просто *решением игры*.

1.2. Приведем без доказательств несколько утверждений из теории игр. Доказательства этих теорем можно прочесть, например, в [12].

Основная теорема теории игр формулируется следующим образом.

Теорема 1.1. *Каждая матричная игра с нулевой суммой имеет решение в смешанных стратегиях, т. е. существуют такие стратегии U^* и W^* первого и второго игрока соответственно, что при любых смешанных стратегиях U и W*

$$M(U, W^*) \leq M(U^*, W^*) \leq M(U^*, W).$$

Теорема 1.2. *Каждой матричной игре с платежной матрицей $A = \|a_{ij}\|$ ($a_{ij} > 0$) соответствует следующая пара двойственных задач линейного программирования:*

$$\left. \begin{aligned} \sum_{j=1}^n x_j &\rightarrow \max, \\ \sum_{j=1}^n a_{ij} x_j &\leq 1, \quad i = 1, 2, \dots, m, \\ x_j &\geq 0, \quad j = 1, 2, \dots, n, \end{aligned} \right\} \quad (1.3)$$

и

$$\left. \begin{aligned} \sum_{i=1}^m y_i &\rightarrow \min, \\ \sum_{i=1}^m a_{ij} y_i &\geq 1, \quad j = 1, 2, \dots, n, \\ y_i &\geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \right\} \quad (1.4)$$

Составляющие оптимальных стратегий u_i^* и w_j^* игры связаны с компонентами y_i^* и x_j^* оптимальных планов двойственных задач линейного программирования следующими формулами:

$$w_j^* = \frac{x_j^*}{\sum_{j=1}^n x_j^*}, \quad u_i^* = \frac{y_i^*}{\sum_{i=1}^m y_i^*}.$$

Цена игры

$$v = \frac{1}{\sum_{j=1}^n x_j^*} = \frac{1}{\sum_{i=1}^m y_i^*}.$$

Вместо приведенной пары двойственных задач (1.3), (1.4) иногда удобнее рассматривать эквивалентную ей пару взаимосопряженных задач линейного программирования

$$\left. \begin{aligned} v &\rightarrow \min, \\ \sum_{j=1}^n a_{ij} \omega_j &\leq v, \quad i = 1, 2, \dots, m, \\ \sum_{j=1}^n \omega_j &= 1, \\ \omega_j &\geq 0, \quad j = 1, 2, \dots, n, \end{aligned} \right\} \quad (1.5)$$

и

$$\left. \begin{aligned} v &\rightarrow \max, \\ \sum_{i=1}^m a_{ij} u_i &\geq v, \quad j = 1, 2, \dots, n, \\ \sum_{i=1}^m u_i &= 1, \\ u_i &\geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \right\} \quad (1.6)$$

Заметим, что сведение матричной игры к эквивалентной ей паре взаимосопряженных задач (1.3), (1.4) требует положительности элементов a_{ij} матрицы игры A . Сведение же игры к паре двойственных задач (1.5), (1.6) не накладывает никаких ограничений на элементы матрицы A . Однако каждая из задач (1.5), (1.6) содержит на одно переменное и одно ограничение больше, чем соответствующая задача (1.3), (1.4).

Теорема 1.2 устанавливает параметры двойственной пары задач линейного программирования, решение которых определяет оптимальные стратегии и цену исходной игры. Можно идти и обратным путем. Каждой паре взаимосопряженных задач линейного программирования можно привести в соответствие матричную игру, цена и оптимальные стратегии которой позволяют вычислить оптимальные планы двойственных задач (если, конечно, они существуют). Следует подчеркнуть, что в то время как матричные игры всегда имеют оптимальные стратегии, задачи линейного программирования могут и не иметь решений.

Рассмотрим пару двойственных задач:

$$\left. \begin{array}{l} CX \rightarrow \max, \\ AX \leq B, \\ X \geq 0 \end{array} \right\} \quad (1.7)$$

и

$$\left. \begin{array}{l} YB \rightarrow \min, \\ YA \geq C, \\ Y \geq 0. \end{array} \right\} \quad (1.8)$$

Построим игру с платежной матрицей

$$P = \left\| \begin{array}{ccc} 0 & A & -B \\ -A^T & 0 & C^T \\ B^T & -C & 0 \end{array} \right\|. \quad (1.9)$$

Имеет место следующее утверждение.

Теорема 1.3. *Пара двойственных задач (1.7), (1.8) имеет решение тогда и только тогда, когда игра с платежной матрицей (1.9) имеет такую оптимальную стратегию $U^* = (u_1^*, \dots, u_{m+n}^*, u_{m+n+1}^*)$, что $u_{m+n+1}^* > 0$.*

При этом

$$y_i^* = \frac{u_i^*}{u_{m+n+1}^*}, \quad i = 1, 2, \dots, m,$$

$$x_j^* = \frac{u_{m+j}^*}{u_{m+n+1}^*}, \quad j = 1, 2, \dots, n.$$

Таким образом, решение пары двойственных задач линейного программирования сведено к вычислению оптимальных стратегий игры с симметричной платежной матрицей P порядка $m+n+1$.

В случае, когда все элементы матрицы условий A неотрицательны и она не содержит нулевых столбцов, а все компоненты векторов B и C положительны, можно построить эквивалентную игру с платежной матрицей из n строк и m столбцов. (Заметим, что задачи линейного программирования, для которых выполнены указанные условия, всегда разрешимы.)

Платежная матрица эквивалентной игры —

$$\tilde{A} = \|\tilde{a}_{ij}\|,$$

где

$$\tilde{a}_{ij} = a_{ij}/b_i c_j.$$

Если $U^* = (u_1^*, \dots, u_m^*)$ — оптимальная стратегия первого игрока, $W^* = (w_1^*, \dots, w_n^*)$ — оптимальная стратегия второго игрока, а $v > 0$ — цена игры, то наборы чисел

$$x_j^* = \frac{w_j^*}{c_j v}, \quad j = 1, 2, \dots, n, \text{ и } y_i^* = \frac{u_i^*}{b_i v}, \quad i = 1, 2, \dots, m,$$

являются решениями задач (1.7) и (1.8) соответственно.

Размеры платежных матриц во втором представлении задач линейного программирования в виде эквивалентной игры меньше, чем в первом представлении. Тем не менее в ряде случаев удобнее пользоваться первым представлением, так как для некоторых вычислительных методов выгоднее иметь игру в симметричной форме.

§ 2. МЕТОДЫ РЕШЕНИЯ ИГР

2.1. Рассмотрим итеративный метод решения игр, предложенный Брауном [3] и обоснованный Дж. Робинсон [36]. Этот очень простой, но недостаточно эффективный метод носит название метода «фиктивной игры» или метода «вилки».

Предположим, что игра с матрицей $A = \|a_{ij}\|_{m,n}$ многократно повторяется, а игроки, запоминая стратегии, выбираемые каждым партнером в каждой партии, оценивают полученные при этом выигрыши и постепенно шлифуют свое мастерство. Можно представить себе следующую схему совершенствования от партии к партии поведения игроков.

В первой партии первый игрок, не имея опыта и знаний, выбирает произвольную чистую стратегию. Пусть ее номер i_1 . Будем рассматривать эту чистую стратегию как смешанную стратегию $U^{(1)}$, в которой содержится только одна ненулевая компонента. Второй игрок, зная выбор партнера, выбирает стратегию, наилучшую против $U^{(1)}$. Другими словами, второй игрок выбирает чистую стратегию с номером j_1 , отвечающим наименьшему числу $a_{i_1 j_1}$ строки i_1 платежной матрицы A . Назовем эту стратегию $W^{(1)}$. Вслед за этим первый игрок, забыв о причинах, побудивших противника выбрать стратегию $W^{(1)}$, выбирает

чистую стратегию i_2 , наилучшую против стратегии j_1 противника. Второй игрок, принимая решение, знает, что его партнер выбрал в одной партии чистую стратегию i_1 , а в другой — чистую стратегию i_2 . Второй игрок предполагает, что его противник придерживается смешанной стратегии $U^{(2)}$, содержащей чистые стратегии i_1 и i_2 , каждую с вероятностью $1/2$. При этом допущении второй игрок должен выбрать чистую стратегию j_2 , при которой его средний проигрыш минимален. Первый игрок, выбирая чистую стратегию в очередной партии, исходит из аналогичных соображений. Выбранная им чистая стратегия i_3 должна быть наилучшей против смешанной стратегии $W^{(2)}$, содержащей чистые стратегии j_1 и j_2 , каждую с вероятностью $1/2$. Стратегия j_3 , которую после этого выбирает второй игрок, должна ему обеспечить минимальный средний проигрыш при смешанной стратегии партнера, содержащей чистые стратегии i_1 , i_2 , i_3 с вероятностью $1/3$ каждая. Процесс последовательного совершенствования стратегий продолжается сколь угодно долго. Пересчитывая после каждой партии частоты появления чистых стратегий противника в ходе всей игры, каждый из игроков принимает их за вероятности, с которыми чистые стратегии входят в смешанные стратегии партнера, и выбирает очередную чистую стратегию наилучшей против данной смешанной стратегии другого игрока.

Описанная простая схема «обучения на опыте» может быть формализована следующим образом.

Пусть после s -й партии первый игрок обнаружил, что его противник выбрал j -ю чистую стратегию s_j раз ($j = 1, 2, \dots, n$). На этом основании первый игрок принимает предположение, что его противник придерживается смешанной стратегии $W^{(s)} = (s_1/s, s_2/s, \dots, s_n/s)$. При этом допущении первый игрок выбирает в $(s+1)$ -й партии чистую стратегию e_{i_s} , дающую максимальный средний выигрыш против $W^{(s)}$. Номер i_s выбранной чистой стратегии есть номер максимальной компоненты вектора $AW^{(s)}$.

Аналогичным образом второй игрок после t партий предполагает, что смешанная стратегия противника имеет вид $U^{(t)} = (t_1/t, \dots, t_m/t)$, где t_i — частота появления i -й стратегии в предыдущих партиях. В $(t+1)$ -й партии

второй игрок выбирает чистую стратегию e_{j_t} , обеспечивающую ему минимальный средний проигрыш. Номер j_t этой чистой стратегии — номер минимальной компоненты вектора $U^{(t)}A$.

Таким образом,

$$U^{(s+1)} = \frac{s}{s+1} U^{(s)} + \frac{1}{s+1} e_{i_s}, \quad (2.1)$$

$$W^{(t+1)} = \frac{t}{t+1} W^{(t)} + \frac{1}{t+1} e_{j_t}, \quad (2.2)$$

где i_s — номер максимальной компоненты вектора $AW^{(s)}$, j_t — номер минимальной составляющей вектора $U^{(t)}A$.

При описании метода фиктивной игры мы предполагали, что оба игрока делают выборы поочередно. В литературе по теории игр чаще рассматривается итеративный процесс, в котором одновременно определяются стратегии обоих игроков. В этой процедуре оба игрока одновременно делают первый выбор и, получая информацию о решении противника, одновременно выбирают в каждой последующей партии чистую стратегию, наиболее выгодную против сформированной до этого смешанной стратегии противника.

Накопленный до сих пор опыт решения игр показывает, что первая процедура (с поочередным выбором стратегий) сходится гораздо быстрее второй процедуры (с одновременным выбором стратегий).

Дж. Робинсон [36] доказала, что после большого числа партий выигрыш в методе фиктивной игры стабилизируется и стремится к цене игры. Теорема Робинсон о сходимости описанного итеративного процесса утверждает, что

$$\lim_{s \rightarrow \infty} \min_j (U^{(s)}A) = \lim_{s \rightarrow \infty} \max_i (AW^{(s)}) = v, \quad (2.3)$$

где v — цена игры.

Мы не будем приводить здесь довольно громоздкого доказательства этого интуитивно естественного утверждения. Отметим только, что итеративный процесс сходится чрезвычайно медленно. Известна следующая оценка [44]. Ошибка в определении цены игры v после s шагов итеративного метода Брауна с одновременным выбором стратегий

не превосходит величины порядка $s^{-1/(m+n-2)}$. Точнее,

$$\left| v - \max_i AW^{(s)} \right| < Cs^{-\frac{1}{m+n-2}}, \quad (2.4)$$

$$\left| v - \min_j U^{(s)} A \right| < Cs^{-\frac{1}{m+n-2}}, \quad (2.5)$$

где $C = \text{const}$.

Пока неясно, является ли приведенная оценка точной. Есть основания ожидать, что ее можно улучшить (см. [44]).

На рис. 9.1 приведены графики зависимости $\delta v^{(s)}$ от s , вычисленные при определении оптимальных стратегий методом фиктивной игры с одновременным и поочередным выбором стратегий для игр с платежными матрицами различных размеров. Элементы платежных матриц выбирались случайным образом. Размеры матриц указаны на рисунке.

Как видим, во всех случаях итеративный процесс сходится гораздо быстрее, чем этого можно было ожидать по оценочным формулам (2.4) и (2.5).

2.2. Рассмотрим итеративный метод численного решения прямоугольных игр, предложенный Дж. фон-Нейманом [31].

Из приведенных в предыдущем параграфе результатов следует, что решение игры с платежной матрицей $\|a_{ij}\|_{m,n}$ сводится к выбору чисел u_i ($i = 1, 2, \dots, m$), w_j ($j = 1, 2, \dots, n$) и v таких, что

$$\sum_{i=1}^m a_{ij} u_i \geq v, \quad j = 1, 2, \dots, n, \quad (2.6)$$

$$\sum_{j=1}^n a_{ij} w_j \leq v, \quad i = 1, 2, \dots, m, \quad (2.7)$$

$$u_i \geq 0, \quad i = 1, 2, \dots, m, \quad (2.8)$$

$$\sum_{i=1}^m u_i = 1, \quad (2.9)$$

$$w_j \geq 0, \quad j = 1, 2, \dots, n, \quad (2.10)$$

$$\sum_{j=1}^n w_j = 1. \quad (2.11)$$

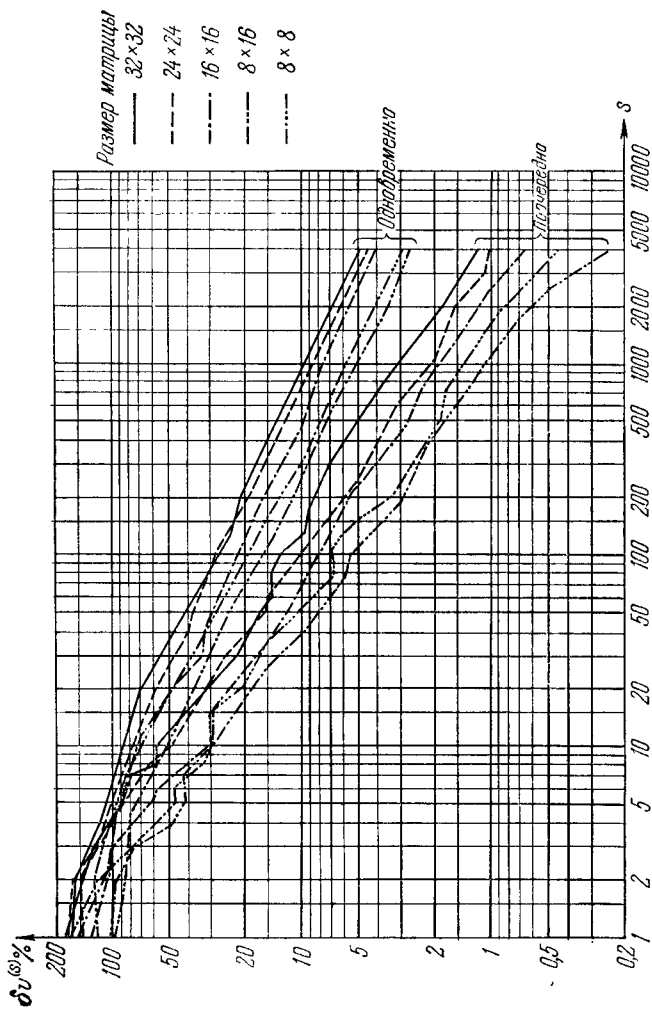


Рис. 9.1.

Введем в рассмотрение невязки

$$P_i = \sum_{j=1}^n a_{ij} \omega_j - v, \quad (2.12)$$

$$Q_j = v - \sum_{i=1}^m a_{ij} u_i, \quad (2.13)$$

которые в свою очередь определяют неотрицательные параметры

$$p_i = \max(P_i, 0), \quad (2.14)$$

$$q_j = \max(Q_j, 0). \quad (2.15)$$

Значения функции

$$\Phi = \sum_{i=1}^m p_i^2 + \sum_{j=1}^n q_j^2 \quad (2.16)$$

оценивают близость выбранных стратегий $U = (u_1, \dots, u_m)$ и $W = (\omega_1, \dots, \omega_n)$ к оптимальным стратегиям и величины v к цене игры. Для вектора (U, W, v) , представляющего решение игры, функция Φ обращается в нуль.

Метод Неймана является итеративным процессом последовательного вычисления векторов U и W и скаляра v , устремляющим Φ к нулю.

В качестве начальных векторов $U^{(0)}$ и $W^{(0)}$ могут быть выбраны произвольные векторы, удовлетворяющие условиям (2.8) — (2.11). Обычно принимают

$$u_i^{(0)} = \frac{1}{m}, \quad \omega_j^{(0)} = \frac{1}{n}, \quad v^{(0)} = \frac{1}{nm} \sum_{i=1}^m \sum_{j=1}^n a_{ij}. \quad (2.17)$$

Переход от s -й итерации к $(s+1)$ -й производится по формулам

$$u_i^{(s+1)} = (1 - \alpha^{(s)}) u_i^{(s)} + \alpha^{(s)} \bar{u}_i^{(s)}, \quad (2.18)$$

$$\omega_j^{(s+1)} = (1 - \alpha^{(s)}) \omega_j^{(s)} + \alpha^{(s)} \bar{\omega}_j^{(s)}, \quad (2.19)$$

$$v^{(s+1)} = (1 - \alpha^{(s)}) v^{(s)} + \alpha^{(s)} \bar{v}^{(s)}, \quad (2.20)$$

где корректуры $\bar{u}_i^{(s)}$ и $\bar{\omega}_j^{(s)}$ стратегий и поправка $\bar{v}^{(s)}$ к величине $v^{(s)}$ вычисляются по следующим формулам:

$$\bar{u}_i^{(s)} = \frac{p_i^{(s)}}{\sum_{i=1}^m p_i^{(s)}}, \quad \bar{\omega}_j^{(s)} = \frac{q_j^{(s)}}{\sum_{j=1}^n q_j^{(s)}}, \quad \bar{v}^{(s)} = \sum_{j=1}^n \sum_{i=1}^m a_{ij} \bar{u}_i^{(s)} \bar{\omega}_j^{(s)}. \quad (2.21)$$

Если $\sum_{i=1}^m p_i^{(s)} \left(\text{или} \sum_{j=1}^n q_j^{(s)} \right)$ окажется равной нулю,

можно выбрать в качестве $\bar{u}_i^{(s)}$ (соответственно $\bar{w}_j^{(s)}$) любые неотрицательные числа, сумма которых равна единице.

Соотношения

$$\sum_{i=1}^m p_i^{(s)} = \sum_{j=1}^n q_j^{(s)} = 0$$

свидетельствуют о том, что решение найдено и процесс вычислений должен быть прекращен.

Величина $\alpha^{(s)}$ — параметр управления, определяющий по формулам (2.18) — (2.20) переход от одной итерации к другой. Выбор $\alpha^{(s)}$ должен обеспечить убывание функции Φ от итерации к итерации.

Сравним значения Φ для двух соседних итераций. Легко непосредственно проверить, что невязки $(P_i^{(s+1)}, Q_j^{(s+1)})$ и $(P_i^{(s)}, Q_j^{(s)})$ в соседних итерациях связаны между собою формулами вида (2.18) — (2.20)

$$P_i^{(s+1)} = (1 - \alpha^{(s)}) P_i^{(s)} + \alpha^{(s)} \bar{P}_i^{(s)},$$

$$Q_j^{(s+1)} = (1 - \alpha^{(s)}) Q_j^{(s)} + \alpha^{(s)} \bar{Q}_j^{(s)},$$

где

$$\bar{P}_i^{(s)} = \sum_{j=1}^n a_{ij} \bar{w}_j^{(s)} - \bar{v}^{(s)}, \quad (2.22)$$

$$\bar{Q}_j^{(s)} = \bar{v}^{(s)} - \sum_{i=1}^m a_{ij} \bar{u}_i^{(s)}. \quad (2.23)$$

Согласно формулам (2.14) $p_i = 0$ при $P_i \leq 0$ и $p_i = P_i$ при $P_i > 0$.

Поэтому всегда

$$[p_i^{(s+1)}]^2 \leq [(1 - \alpha^{(s)}) p_i^{(s)} + \alpha^{(s)} \bar{P}_i^{(s)}]^2,$$

и аналогично

$$[q_j^{(s+1)}]^2 \leq [(1 - \alpha^{(s)}) q_j^{(s)} + \alpha^{(s)} \bar{Q}_j^{(s)}]^2.$$

Полученные неравенства позволяют сравнить значения $\Phi^{(s+1)}$ и $\Phi^{(s)}$. Имеем

$$\begin{aligned} \Phi^{(s+1)} = & \sum_{i=1}^m [p_i^{(s+1)}]^2 + \sum_{j=1}^n [q_j^{(s+1)}]^2 \leq [1 - \alpha^{(s)}]^2 \left(\sum_{i=1}^m [p_i^{(s)}]^2 + \right. \\ & \left. + \sum_{j=1}^n [q_j^{(s)}]^2 \right) + 2\alpha^{(s)}(1 - \alpha^{(s)}) \left(\sum_{i=1}^m p_i^{(s)} \bar{p}_i^{(s)} + \sum_{j=1}^n q_j^{(s)} \bar{q}_j^{(s)} \right) + \\ & + [\alpha^{(s)}]^2 \left(\sum_{i=1}^m [\bar{p}_i^{(s)}]^2 + \sum_{j=1}^n [\bar{q}_j^{(s)}]^2 \right). \end{aligned}$$

Из формул (2.21) — (2.23) вытекает, что

$$\sum_{i=1}^m p_i^{(s)} \bar{p}_i^{(s)} + \sum_{j=1}^n q_j^{(s)} \bar{q}_j^{(s)} = 0.$$

Следовательно,

$$\Phi^{(s+1)} \leq [1 - \alpha^{(s)}]^2 \Phi^{(s)} + [\alpha^{(s)}]^2 C^{(s)}, \quad (2.24)$$

где

$$C^{(s)} = \sum_{i=1}^m [\bar{p}_i^{(s)}]^2 + \sum_{j=1}^n [\bar{q}_j^{(s)}]^2. \quad (2.25)$$

Правая часть неравенства (2.24) достигает наименьшего значения при

$$\alpha^{(s)} = \frac{\Phi^{(s)}}{\Phi^{(s)} + C^{(s)}}. \quad (2.26)$$

Заметим, что $\Phi^{(s)} \geq 0$ и $C^{(s)} \geq 0$. Поэтому $0 \leq \alpha^{(s)} \leq 1$.

Неравенство (2.24) при $\alpha^{(s)}$ из (2.26) принимает вид

$$\Phi^{(s+1)} \leq \frac{\Phi^{(s)} C^{(s)}}{\Phi^{(s)} + C^{(s)}}, \quad \text{или} \quad \frac{1}{\Phi^{(s+1)}} \geq \frac{1}{\Phi^{(s)}} + \frac{1}{C^{(s)}}. \quad (2.27)$$

Оценим сходимость метода Неймана. Имеет место следующее утверждение.

Теорема 2.1. *Итеративный метод Неймана сходится. При этом $\Phi^{(s)}$ стремится к нулю, как $1/s$.*

Доказательство. Пусть

$$\lambda = \max_{i,j} a_{ij}, \quad \mu = \min_{i,j} a_{ij}.$$

Очевидно,

$$\lambda \geq \sum_{j=1}^n a_{ij} \bar{w}_j^{(s)} \geq \mu, \quad \lambda \geq \sum_{i=1}^m a_{ij} \bar{u}_i^{(s)} \geq \mu, \quad \lambda \geq \bar{v}^{(s)} \geq \mu.$$

Отсюда

$$\lambda - \mu \geq \sum_{j=1}^n a_{ij} \bar{w}_j^{(s)} - \bar{v}^{(s)} = \bar{P}_i^{(s)} \geq \mu - \lambda,$$

$$\lambda - \mu \geq - \sum_{i=1}^m a_{ij} \bar{u}_i^{(s)} + \bar{v}^{(s)} = \bar{Q}_j^{(s)} \geq \mu - \lambda$$

или

$$\lambda - \mu \geq |\bar{Q}_j^{(s)}|, \quad \lambda - \mu \geq |\bar{P}_i^{(s)}|.$$

Следовательно (см. (2.25)),

$$C^{(s)} = \sum_{i=1}^m [\bar{P}_i^{(s)}]^2 + \sum_{j=1}^n [\bar{Q}_j^{(s)}]^2 \leq (m+n)(\lambda - \mu)^2.$$

При этом условии неравенство (2.27) принимает вид

$$\frac{1}{\Phi^{(s+1)}} \geq \frac{1}{\Phi^{(s)}} + \frac{1}{(m+n)(\lambda - \mu)^2}.$$

Полученная система рекуррентных неравенств приводит к соотношению

$$\frac{1}{\Phi^{(s+1)}} \geq \frac{1}{\Phi^{(0)}} + \frac{s+1}{(m+n)(\lambda - \mu)^2} > \frac{s+1}{(m+n)(\lambda - \mu)^2},$$

откуда

$$\Phi^{(s)} < \frac{a}{s},$$

где $a = (m+n)(\lambda - \mu)^2$, что и следовало доказать.

Расчет по методу Неймана не вызывает затруднений. Блок-схема вычислений представлена на рис. 9.2.

На схеме приняты следующие обозначения:

$$p = \sum_{i=1}^m p_i, \quad q = \sum_{j=1}^n q_j, \quad S_j = \sum_{i=1}^m a_{ij} \bar{u}_i, \quad T_i = \sum_{j=1}^n a_{ij} \bar{w}_j.$$

Следуя блок-схеме, изображенной на рис. 9.2, можно оценить число операций на каждом шаге метода. Имеем:

число сравнений (вычисление $\max(0, P_i)$ и $\max(0, Q_j)$) — $-(m+n)$;

число сложений и вычитаний — $[2mn + 7(m+n) + (m \text{ или } n) + 4]$;

число умножений — $[2mn + 4(m+n) + (m \text{ или } n) + 1]$;
число делений — $(m+n+1)$.

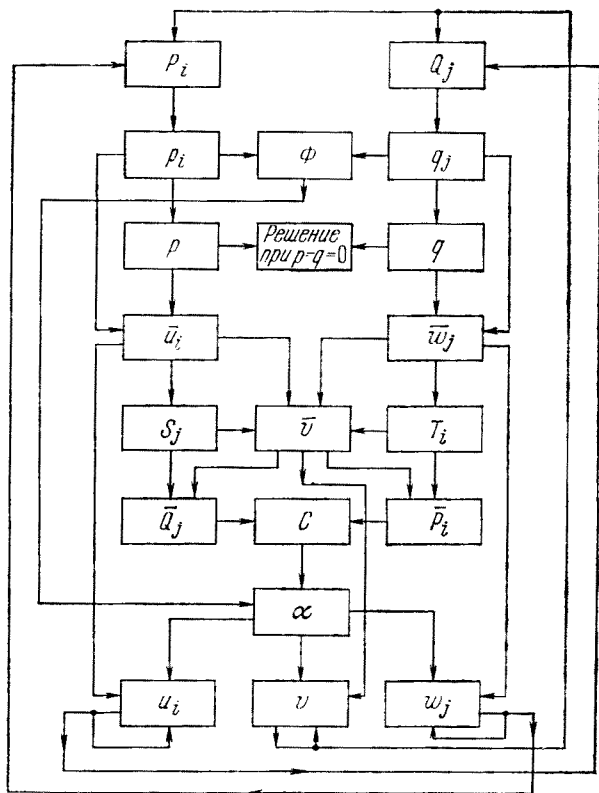


Рис. 9.2.

2.3. Отдельные шаги итеративных методов обычно не трудоемки. Однако сходимость этих методов весьма медленная. Поэтому при необходимости получить относительно точное решение игры или задачи линейного программирования целесообразно отдавать предпочтение конечным методам.

Известны различные попытки ускорить сходимость итеративных методов. Интуитивно представляется, что можно ускорить сходимость метода фиктивной игры, если после некоторого числа итераций отказаться от учета устаревшего опыта и определять очередную стратегию только по фиксированному числу последних выборов противника. Однако анализ этой модификации показывает, что метод фиктивной игры с конечной памятью не гарантирует сходимости итеративного процесса. Можно построить примеры, когда ограничение памяти приводит к неустойчивости процесса.

В общем случае оказывается бесполезной и другая интуитивно приемлемая модификация метода фиктивной игры, в которой предшествующий опыт учитывается с весом, убывающим по мере его старения (случай затухающей памяти).

Можно, однако, опираясь на результаты В. А. Волконского [6], высказать следующее предположение.

Сходимость метода фиктивной игры может быть, по видимому, обеспечена, а возможно, и ускорена, если заменить формулы (2.1), (2.2) для рекуррентного формирования стратегий противников формулами вида

$$U^{(s+1)} = (1 - \lambda_{s+1}) U^{(s)} + \lambda_{s+1} e_{i_s}, \quad (2.28)$$

$$W^{(t+1)} = (1 - \lambda_{t+1}) W^{(t)} + \lambda_{t+1} e_{j_t}, \quad (2.29)$$

где

$$\lim_{s \rightarrow \infty} \lambda_s = 0, \quad \sum_{s=1}^{\infty} \lambda_s = \infty, \quad 0 \leq \lambda_s \leq 1. \quad (2.30)$$

В формулах (2.1), (2.2) $\lambda_s = 1/s$ удовлетворяет условиям (2.30). Если весовые коэффициенты убывают с ростом s , как $e^{-\alpha s}$, $\alpha > 0$, второе из условий (2.30) не выполняется и соответствующий процесс решения игры, вообще говоря, не сходится.

Вряд ли следует ожидать, что можно заранее выбрать фиксированную последовательность λ_s , которая при любой платежной матрице обеспечит существенное ускорение сходимости по сравнению с методом Брауна. Представляется, что сходимость может быть ускорена, если формировать весовые коэффициенты λ_s в процессе решения

игры. Один из возможных путей формирования коэффициентов λ_s предложен В. В. Амвросиенко [1].

Рекомендации Амвросиенко основаны на значительном опыте решения игр методом фиктивной игры.

Замечено, что в итеративном процессе метода фиктивной игры противники применяют некоторое число раз подряд одни и те же стратегии. С увеличением номера s итерации число повторений одной и той же стратегии растет. При одновременном выборе стратегий обоими противниками повторения начинаются уже с малых s , а при поочередном выборе — с относительно больших s . Можно вычислить количество последовательных повторений игроками одних и тех же ходов и заменить соответствующее число итераций одной итерацией. Тогда на каждой итерации по крайней мере один из противников будет применять стратегию, отличную от используемой на предыдущем шаге. Общее число итераций, требуемых для достижения заданной точности решения игры, при этом существенно сократится.

При одновременном выборе стратегий обоими игроками число l последовательных повторений (длина серии повторяющихся стратегий) после s -й итерации определяется меньшим из чисел l'_i и l''_j , где

$$l'_i = \begin{cases} 1 + \left[\frac{s(AW^{(s)})_{k''} - s(AW^{(s)})_i}{a_{ik'} - a_{k''k'}} \right] & \text{при } a_{ik'} > a_{k''k'}, \\ \infty & \text{при } a_{ik'} \leq a_{k''k'}, \end{cases}$$

$$l''_j = \begin{cases} 1 + \left[\frac{s(U^{(s)}A)_j - s(U^{(s)}A)_{k'}}{a_{k''k'} - a_{k''j}} \right] & \text{при } a_{k''j} < a_{k''k'}, \\ \infty & \text{при } a_{k''j} \geq a_{k''k'}, \end{cases}$$

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

Здесь k'' — номер максимального элемента столбца $AW^{(s)}$, k' — номер минимального элемента строки $U^{(s)}A$, a_{ij} — элементы исходной матрицы выигрышей, $[M]$ — целая часть числа M .

Все l итераций могут быть заменены одной. При этом следует в схеме метода фиктивной игры добавить к строке $sU^{(s)}A$ умноженную на l строку матрицы выигрышей, характеризующую повторяющуюся стратегию первого

игрока, а к столбцу $sAW^{(s)}$ прибавить умноженный на l столбец исходной платежной матрицы, отвечающий повторяющейся стратегии второго игрока. Соответствующим образом учитывая l -разовое применение чистых стратегий, следует изменить смешанные стратегии $U^{(s)}$ и $W^{(s)}$.

Модификация В. В. Амвросиенко соответствует методу

Брауна с весовыми коэффициентами вида $\lambda_s = l_s / \sum_{i=1}^s l_i$, где

l_s — число «склеенных» итераций метода фиктивной игры на s -м шаге модифицированного метода.

В. В. Амвросиенко получил следующую эмпирическую оценку отклонения ε выигрышей игроков после s шагов модифицированной «фиктивной игры», в зависимости от размера n квадратной платежной матрицы, элементы которой равномерно распределены на отрезке $[0, 1]$:

$$\varepsilon \leq 0,003 \frac{n^{2,26}}{s}.$$

Возможны также модификации, ускоряющие сходимость итеративного метода фон Неймана. Сравнивая предложенный им метод с конечными методами линейного программирования, Дж. фон Нейман отмечает, что, несмотря на то, что априорные оценки метода последовательного улучшения плана менее благоприятны, чем оценка сходимости итеративного метода, опыт заставляет отдавать предпочтение конечным методам линейного программирования. Однако имеются основания ожидать, что итеративный процесс фон Неймана можно ускорить различными приемами, которые сводятся к сглаживанию итеративной последовательности с некоторой конечной памятью.

§ 3. МЕТОДЫ «ШТРАФНЫХ ФУНКЦИЙ»

3.1. Помимо итеративных методов линейного программирования, полученных преобразованием соответствующих методов решения игр, имеются итеративные методы, разработанные непосредственно для вычисления оптимальных планов линейных экстремальных задач. Такие методы рассмотрены в [47, 48, 38, 4, 35] и других работах. В некоторых из них, например в

методе Удзавы [38] и методе Булавского [4] итеративный процесс требует включения в оптимизируемую функцию «небольшой» нелинейности и решения специально разработанными для этой цели методами полученных вспомогательных задач квадратичного программирования. Мы опускаем здесь изложение этих методов, требующее ссылки на основы теории выпуклого программирования.

В методе Томаша Петшиковского [34] задача линейного программирования приводится к эквивалентной безусловной выпуклой экстремальной задаче, которая может быть решена градиентным методом. Некоторый опыт решения задач линейного программирования по методу Петшиковского, накопленный в ВЦ АН СССР, свидетельствует об эффективности метода.

Запишем задачу линейного программирования в следующем удобном для излагаемого метода виде:

$$L = (C, X) \rightarrow \max, \quad (3.1)$$

$$\varphi_i(X) = (A^{(i)}, X) - b_i \geq 0, \quad i = 1, 2, \dots, m, \quad (3.2)$$

$$\varphi_{m+j}(X) = x_j = (e_j, X) \geq 0, \quad j = 1, 2, \dots, n_1 \leq n. \quad (3.3)$$

Здесь $A^{(i)}$ — i -я вектор-строка матрицы условий, e_j — n -мерный единичный вектор с единицей на j -м месте.

Полагая

$$A^{(m+i)} = e_i, \quad b_{m+i} = 0, \quad i = 1, 2, \dots, n_1,$$

перепишем условия (3.2) — (3.3) в эквивалентной форме:

$$\varphi_i(X) = (A^{(i)}, X) - b_i \geq 0, \quad i = 1, 2, \dots, m + n_1. \quad (3.4)$$

Будем в этом и следующем пунктах считать, что $C \neq 0$ и область определения линейной формы ограничена. Назовем многогранное множество, определяемое условиями (3.4), многогранником W .

Рассмотрим функционал

$$G_\mu(X) = (C, X) + \frac{\mu}{2} \sum_{i=1}^{m+n_1} \tilde{\delta}_i[\varphi_i(X)] [\varphi_i(X)]^2, \quad (3.5)$$

где

$$\tilde{\delta}_i[\varphi_i(X)] = \delta_i(X) = \begin{cases} 0, & \text{если точка } X \text{ удовлетворяет} \\ & i\text{-му условию задачи,} \\ -1, & \text{если точка } X \text{ не удовлетво-} \\ & \text{ряет } i\text{-му условию задачи.} \end{cases}$$

Для условий задачи, записанных в форме (3.4),

$$\delta_i(X) = \begin{cases} 0 & \text{при } \varphi_i(X) \geq 0, \\ -1 & \text{при } \varphi_i(X) < 0. \end{cases}$$

Как видим, функционал $G_\mu(X)$ совпадает с линейной формой (3.1) в области ее определения (3.4). Вне многогранника W функционал $G_\mu(X)$ отличается от линейной формы (C, X) на квадратичную форму

$$\frac{\mu}{2} \sum_{i \in I} [(A^{(i)}, X) - b_i]^2,$$

где I — множество номеров нарушенных ограничений. График $G_\mu(X)$ при различных μ для одномерного случая приведен на рис. 9.3. В качестве

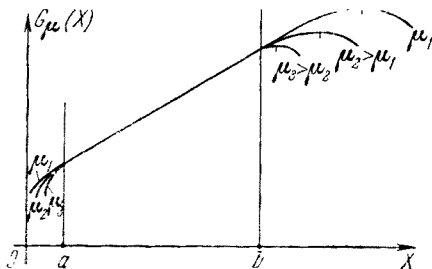


Рис. 9.3.

в области определения линейной формы здесь принят отрезок $[a, b]$.

Ясно, и это наглядно подтверждается графиком, что, вообще говоря, множество M решений задачи (3.1)—(3.3) не совпадает с множеством M_μ точек, в которых функционал G_μ достигает максимума на всем пространстве E_n . Но по мере увеличения μ множество M_μ приближается к множеству M .

Обозначим через $\rho(X, M)$ нижнюю грань расстояний между X и $Y \in M$ по всем точкам Y множества M .

В [34] доказано следующее утверждение.

Теорема 3.1. Для любого $\varepsilon > 0$ существует такое $\mu_0 > 0$, что при любом $\mu > \mu_0$ неравенство

$$\rho(X, M) < \varepsilon$$

выполняется для любого $X \in M_\mu$.

Из теоремы 3.1 следует, что решение задачи линейного программирования (в пределах заданной точности) сводится к вычислению безусловного экстремума функционала $G_\mu(X)$. Полученную задачу можно решать градиентным методом.

Определим последовательность $\{Z^{(k)}\}$ следующим рекуррентным соотношением:

$$Z^{(k+1)} = Z^{(k)} + \theta g^{(k)}, \quad (3.6)$$

где $Z^{(0)}$ — произвольная точка из E_n ; $g^{(k)}$ — градиент функции $G_\mu(X)$ в точке $Z^{(k)}$.

В качестве длины шага $\theta^{(k)}$ градиентного метода выбирается значение θ , обращающее в максимум функцию

$$\zeta(\theta) = G_\mu(Z^{(k)} + \theta g^{(k)}). \quad (3.7)$$

Функция $\zeta(\theta)$ выгнута вверх. Поэтому величина шага $\theta^{(k)}$ — единственный корень уравнения

$$\zeta'(\theta) = 0.$$

Дифференцируя (3.7), имеем

$$\begin{aligned} \zeta'(\theta) &= \frac{\partial}{\partial \theta} G_\mu(Z^{(k)} + \theta g^{(k)}) = \\ &= (C, g^{(k)}) + \mu \sum_{i=1}^{m+n_1} \delta_i(Z^{(k)} + \theta g^{(k)}) [\varphi_i(Z^{(k)}) + \\ &\quad + \theta(A^{(i)}, g^{(k)})] (A^{(i)}, g^{(k)}). \end{aligned}$$

Легко видеть, что $\zeta'(\theta)$ — кусочно-линейная функция переменной θ .

Чтобы вычислить корень уравнения $\zeta'(\theta) = 0$, целесообразно расположить в порядке возрастания точки

$$\theta_i^{(k)} = - \frac{\varphi_i(Z^{(k)})}{(A^{(i)}, g^{(k)})},$$

в которых функция $\zeta'(\theta)$ терпит излом. Точка $\theta^{(k)}$ принадлежит отрезку $[\theta_l^{(k)}, \theta_{l+1}^{(k)}]$, для которого $\zeta'(\theta_l^{(k)}) > 0$, а $\zeta'(\theta_{l+1}^{(k)}) \leq 0$,

$$\theta^{(k)} = \frac{\zeta'(\theta_l^{(k)}) \theta_{l+1}^{(k)} - \zeta'(\theta_{l+1}^{(k)}) \theta_l^{(k)}}{\zeta'(\theta_l^{(k)}) - \zeta'(\theta_{l+1}^{(k)})}.$$

Приведенный выбор шага гарантирует сходимость итеративного процесса.

Всюду, где необходимо подчеркнуть, что последовательность $\{Z^{(k)}\}$ соответствует фиксированному значению параметра μ , будем обозначать ее через $\{Z_\mu^{(k)}\}$.

Имеет место следующее утверждение [35].

Теорема 3.2. Последовательность $\{Z_\mu^{(k)}\}$, определенная соотношением (3.6), при любом $Z^{(0)} \in E_n$ и $\mu > 0$ обладает тем свойством, что

$$\lim_{k \rightarrow \infty} \rho(Z_\mu^{(k)}, M_\mu) = 0.$$

Согласно теореме 3.2 при фиксированных $\mu > 0$ и больших k элементы последовательности $\{Z_\mu^{(k)}\}$ подходят достаточно близко к точкам множества M_μ .

По теореме 3.1 при достаточно больших значениях μ любая точка M_μ близка к множеству решений задачи линейного программирования. Отсюда вытекает, что при достаточно больших μ и k можно подойти сколь угодно близко к решению задачи (3.1)—(3.3).

Опишем теперь вычислительную схему, реализующую процесс построения итеративной последовательности $\{Z^{(k)}\}$.

1. Вычисляем

$$\beta_i^{(k)} = \varphi_i(Z^{(k)}), \quad i = 1, 2, \dots, m + n_1.$$

2. Определяем составляющие градиента $g^{(k)}$

$$g^{(k)} = C + \mu \sum_{i=1}^{m+n_1} \delta_i(Z^{(k)}) \beta_i^{(k)} A^{(i)}.$$

3. Если все составляющие $g_j^{(k)}$ градиента $g^{(k)}$ не превышают по абсолютной величине заданного малого числа $\varepsilon > 0$, то $Z^{(k+1)} \approx Z^{(k)}$, и процесс решения задачи для принятого значения параметра μ закончен.

При $|g_j^{(k)}| > \varepsilon$ определяем $\xi_i^{(k)} = (A^{(i)}, g^{(k)})$.

4. Вычисляем отношения

$$\vartheta_i^{(k)} = -\frac{\beta_i^{(k)}}{\xi_i^{(k)}}, \quad i = 1, 2, \dots, m + n_1, \quad \xi_i^{(k)} \neq 0,$$

и располагаем их в порядке возрастания:

$$\vartheta_{i_1}^{(k)} \leq \vartheta_{i_2}^{(k)} \leq \dots \leq \vartheta_{i_s}^{(k)}.$$

5. Определяем значения функции

$$\begin{aligned} \xi'(\theta) &= \frac{\partial}{\partial \theta} G_\mu(Z^{(k)} + \theta g^{(k)}) = \\ &= (C, g^{(k)}) + \mu \sum_{i=1}^{m+n_1} \delta_i(Z^{(k)} + \theta g^{(k)}) (\beta_i^{(k)} + \theta \xi_i^{(k)}) \xi_i^{(k)} \end{aligned}$$

в точках $\vartheta_{i_1}^{(k)}$.

6. Вычисляем длину шага $\theta^{(k)}$

$$\theta^{(k)} = \frac{\zeta'(\vartheta_{i_l}^{(k)}) \vartheta_{i_{l+1}}^{(k)} - \zeta'(\vartheta_{i_{l+1}}^{(k)}) \vartheta_{i_l}^{(k)}}{\zeta'(\vartheta_{i_l}^{(k)}) - \zeta'(\vartheta_{i_{l+1}}^{(k)})},$$

где $\vartheta_{i_l}^{(k)}$ и $\vartheta_{i_{l+1}}^{(k)}$ определяют отрезок, на котором $\zeta'(\theta)$ меняет знак, т. е. $\zeta'(\vartheta_{i_l}^{(k)}) > 0$, но $\zeta'(\vartheta_{i_{l+1}}^{(k)}) \leq 0$.

7. Строим

$$Z^{(k+1)} = Z^{(k)} + \theta^{(k)} g^{(k)}.$$

Нетрудно проверить, что значения $\beta_i^{(k)}$ можно вычислять по следующим рекуррентным формулам:

$$\beta_i^{(k+1)} = \beta_i^{(k)} + \theta^{(k)} \xi_i^{(k)}.$$

При решении задачи линейного программирования излагаемым в настоящем пункте методом целесообразно начинать вычисления с относительно небольшого значения μ . При больших μ величина $\rho(Z_\mu^{(k)}, M_\mu)$ медленно стремится к нулю. Если в силу условий $|g_j^k| \leq \varepsilon$ итеративный процесс прекратился на векторе $Z_\mu^{(k_0)}$, которому соответствуют большие невязки $\beta_i^{(k_0)} = \varphi_i(Z_\mu^{(k_0)})$ ограничений задачи, следует увеличить μ и повторить процесс построения экстремума функционала $G_\mu(X)$. Процесс решения задачи заканчивается при таком μ и таком k , при которых одновременно выполняются условия

$$\begin{aligned} |g_j^k| &\leq \varepsilon, & j &= 1, 2, \dots, n, \\ -\varphi_i(Z_\mu^{(k)}) &\leq \varepsilon_1, & i &= 1, 2, \dots, m + n_1, \end{aligned}$$

где $\varepsilon > 0$ и $\varepsilon_1 > 0$ — заданные малые числа, определяющие допустимые приближения к оптимальному плану задачи линейного программирования.

3.2. В методе Петшиковского решение условной экстремальной задачи сводится к решению безусловной задачи. Выход точки за допустимую область карается штрафом, сокращающим величину максимизируемой линейной формы.

Штраф за невязки позволяет использовать методы решения безусловных экстремальных задач для вычисления

максимума функционала в некоторой фиксированной области. Меняя зависимость штрафа от величины невязки, можно получить различные методы решения задач математического программирования. В методе Петшиковского зависимость штрафа Φ от величины невязки φ представляется функцией

$$\Phi(\varphi) = \sum_{i=1}^{m+n_1} \Phi_i(\varphi_i) = \frac{\mu}{2} \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i) [\varphi_i(X)]^2. \quad (3.8)$$

Обоснование сходимости метода не претерпит существенных изменений по сравнению с доказательством сходимости метода Петшиковского, если заменить функции штрафа Φ_i вида (3.8) любыми другими гладкими функциями, монотонно убывающими с ростом невязки φ_i и равными нулю при соблюдении условий задачи.

В п. 3.1 рассматривался дискретный вариант метода штрафных функций, приспособленный к реализации на ЦВМ. Нетрудно построить непрерывный аналог этого метода, приспособленный к реализации на аналоговых вычислительных устройствах.

Точка $X(t)$, удовлетворяющая векторному дифференциальному уравнению

$$\frac{dX}{dt} = \text{grad } G_\mu(X), \quad (3.9)$$

будет по мере увеличения t приближаться к точке максимума $G_\mu(X)$. При достаточно большом μ точка максимума $G_\mu(X)$ окажется сколь угодно близкой к решению соответствующей задачи линейного программирования.

Для метода Петшиковского

$$\frac{dX}{dt} = C + \mu \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i) \varphi_i(X) A^{(i)}, \quad (3.10)$$

где, как и ранее, вектор $A^{(i)} = (a_{i1}, \dots, a_{in})$ нормален к гиперплоскости, определяемой i -м условием системы (3.4), и направлен внутрь области определения задачи.

Меняя функции штрафа, можно получить из схемы (3.9) различные итеративные методы решения задачи линейного программирования. Доказательство сходимости этих методов при упомянутых ограничениях на выбор

функции штрафа в принципе не отличается от обоснования сходимости дискретных реализаций метода Петшиковского.

Все приведенные рассуждения могут быть распространены и на случай выпуклого программирования.

Пайн [33] предложил метод линейного программирования, укладывающийся в схему (3.9). Однако функция штрафа, выбранная Пайном, не является гладкой функцией. В этом методе

$$\Phi_i(\varphi_i) = \tilde{\delta}_i(\varphi_i) \varphi_i(X). \quad (3.11)$$

Дифференциальное уравнение (3.9) при функции штрафа вида (3.11) принимает вид

$$\frac{dX}{dt} = C + \mu \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i) A^{(i)}. \quad (3.12)$$

Компоненты вектора $X(t)$ удовлетворяют системе дифференциальных уравнений

$$\frac{dx_j}{dt} = c_j + \mu \sum_{i=1}^{m+n_1} a_{ij} \delta_i(X). \quad (3.13)$$

Производные функции штрафа в методе Пайна — разрывные функции. Поэтому сходимость метода Пайна не вытекает из соответствующих рассуждений для непрерывных аналогов метода Петшиковского с гладкими функциями штрафа. Условия сходимости метода Пайна должны быть установлены самостоятельно.

Для решения системы дифференциальных уравнений (3.9) могут быть использованы аналоговые моделирующие устройства.

Возможность автоматизации решения задач линейного (и не только линейного) программирования может быть существенно расширена при использовании схем, включающих аналоговые устройства и ЦВМ. Некоторые рекомендации в этом направлении обсуждаются в [7].

3.3. Численный анализ показывает, что скорость приближения величины линейной формы к оптимуму в итеративных методах линейного программирования (особенно

в тех, которые получены преобразованием соответствующих методов решения игр) существенно убывает по мере увеличения числа итераций. Другими словами, опыт вычислений показывает, что за относительно небольшое число шагов итеративные методы позволяют получить некоторое приближение к оптимальному значению линейной формы, уточнение которого происходит очень медленно.

Отсюда два качественных вывода:

1. Использование итеративных методов решения игр для вычисления оптимального плана задачи линейного программирования целесообразно главным образом в тех случаях, когда требуется достаточно быстро получить грубое приближение к решению задачи.

2. При решении игр или задач линейного программирования большого размера могут оказаться целесообразными комбинированные методы, использующие достоинства итеративных и конечных методов. По приближенному решению игры, полученному итеративным путем за относительно небольшое число шагов, определяются планы соответствующей пары двойственных задач линейного программирования.

Градиентный метод, изложенный в [51] (гл. 9, § 5), позволяет перейти от каждого из этих планов к опорному плану соответствующей задачи линейного программирования. Следует ожидать, что выбор вычисленного таким образом опорного плана в качестве начального опорного плана позволит сэкономить на числе шагов, необходимых для решения задачи при помощи одного из конечных методов линейного программирования.

В заключение отметим три важных достоинства итеративных методов, которые заставляют в ряде случаев отдавать им предпочтение по сравнению с конечными методами линейного программирования.

1. Вычислительные схемы, соответствующие итеративным методам, существенно более помехоустойчивы (менее чувствительны к случайным сбоям и к ошибкам вычислений), чем алгоритмы, отвечающие конечным методам линейного программирования.

2. Итеративные методы в большей мере, чем конечные, соответствуют механизмам установления равновесия

в технических, биологических, экономических и других динамических системах. Поэтому итеративные методы решения задач математического программирования получают более естественную интерпретацию в терминах конкретных прикладных задач.

3. При использовании итеративных методов для решения задач линейного программирования можно хранить расширенную матрицу условий задачи в компактном виде (запоминаются лишь ненулевые элементы и номера их позиций). Это весьма важно для решения задач большого размера со слабо заполненными матрицами условий.

1. Амвросиев В. В., Ускорение сходимости метода Брауна решения матричных игр. Экономика и математические методы, № 4, 1965.
2. Бил (Beale E. M. L.), Cycling in the dual simplex algorithm. Nav. Res. Logist. Quart. 2 (1955), 269—275.
3. Браун (Brown G. W.), Iterative solutions of games by fictitious play, Activity Analysis of Production and Allocation, ed. by Koopmans, Cowles Commission for Research in Economics Monograph, № 13, Wiley, New York, 1951, 374—376.
4. Булавский В. А., Итеративный метод решения общей задачи линейного программирования. Сб. «Численные методы оптимального планирования», Экономико-математическая серия, вып. 1, Изд-во Сиб. отд. АН СССР, Новосибирск, 1962, стр. 35—64.
5. Вагнер (Wagner H. M.), A comparison of the original and the revised simplex method, Oper. Res. 5, 3 (1957), 361—369.
6. Волконский В. А., Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции). Экономика и математические методы, № 2, 1965.
7. Гальперин М. В., Короткевич Г. П., Минский И. Н., Рыбасов В. И., К решению задач нелинейного программирования с одним и многими экстремумами на аналоговых вычислительных устройствах, Техническая кибернетика, № 4, 1964.
8. Герчук Я. П., Проблемы оптимального планирования, гл. VI и VII. Экономиздат, 1961.
9. Гольштейн Е. Г., О возможности расширения применимости частных методов линейного программирования. Сб. «Планирование и экономико-математические методы» (К семидесятилетию академика В. С. Немчинова), М., «Наука», 1964, стр. 409—423.
10. Гольштейн Е. Г., Об одном видоизменении метода последовательного улучшения плана. Экономика и математические методы, № 1 (1965), 83—86.
11. Гольштейн Е. Г., Юдин Д. Б., Об одном классе задач планирования народного хозяйства. Проблемы кибернетики, вып. 5 (1961), 165—182.
12. Гольштейн Е. Г., Юдин Д. Б., Новые направления в линейном программировании. Изд-во «Советское радио», 1966.
13. Гофман (Hoffman A. J.), Cycling in the simplex algorithm, National Bureau of Standards Report, December 16, № 2974, 1953.

14. Данциг (Dantzig G. B.), A proof of the equivalence of the programming and the game problem. *Activity Analysis of Production and Allocation*, ed. by Koopmans, T. C., Cowles Commission Monograph, № 13, New York, Wiley, 1951, p. 330—335.

15. Данциг, Орчард-Хейс (Dantzig G. B., Orchard-Hays W.), The product form for the inverse in the simplex method. *Math. Tables Aids Comp.* 8 (1954), 64—67.

16. Данциг, Форд, Фулдерсон (Dantzig G. B., Ford L. R., Fulderson D. R.). Алгоритм для одновременного решения прямой и двойственной задач линейного программирования. Сб. статей «Линейные неравенства и сложные вопросы» под ред. Куна и Таккера, 1959.

17. Зуховицкий С. И., Авдеева Л. И., Линейное и выпуклое программирование. Изд. 2-е, перераб. и дополи., М., «Наука», 1967.

18. Канторович Л. В., О методах анализа некоторых экстремальных планово-производственных задач. *ДАН СССР* 115, № 3 (1957), 441—444.

19. Канторович Л. В., О перемещении масс. *ДАН СССР* 37, № 7—8 (1942), 227—229.

20. Канторович Л. В., Экономический расчет наилучшего использования ресурсов, Изд-во АН СССР, 1959.

21. Канторович Л. В., Гавурии М. К., Применение математических методов в вопросах анализа грузопотоков. Сб. «Проблемы повышения эффективности работы транспорта», Изд-во АН СССР, 1949, стр. 110—138.

22. Канторович Л. В., Залгаллер В. А., Расчет рационального раскроя промышленных материалов. Госстатиздат, 1951, стр. 198.

23. Кауфман (Kaufmann A.), *Methodes et modeles de la recherche opérationnelle*. Paris, Dunod, гл. 2, 1962.

24. Коротков С. Ф., Применение методов линейного и нелинейного программирования к некоторым экономическим задачам рациональной разработки обводненных месторождений. Материалы к конференции по опыту и перспективам применения математических методов и ЭВМ в планировании, АН СССР, Сибирское отд., 1962.

25. Куи (Kuhn H. W.), Венгерский метод решения задачи о назначениях. Сб. «Методы и алгоритмы решения транспортной задачи», Госстатиздат, 1963, стр. 35—52.

26. Лемке (Lemke G. E.), Двойственный метод решения задач линейного программирования. Сб. «Методы решения общей задачи линейного программирования», Госстатиздат, стр. 55—70.

27. Леннон (Lennon B. C.), *Operational Research: its application to the Oil Industry*. *Petroleum* 17, № 4 (1954), 130—133.

28. Леонтьев (Leontieff W. W.), Исследование структуры американской экономики. Госстатиздат, 1958.

29. Манкрес (Munkres J.), Алгоритмы решения задачи выбора и транспортной задачи. Сб. «Методы и алгоритмы решения транспортной задачи», Госстатиздат, 1963, стр. 73—79.

30. Мухаметзянов Ф. М., Салехов Г. С., Чугунов В. Д., Применение линейного программирования к решению

некоторых задач рациональной разработки нефтяных месторождений. Нефть и газ, № 9, 1960.

31. Нейман фон (von Neumann J.), A numerical method to determine optimum strategy. Nav. Res. Logist. Quart. 1, № 2 (1954), 109—115.

32. Николаевский Н. М., Бучин А. Н., О минимальном дебите нефтяной скважины. Труды ВНИИ, вып. XXVI, 1960.

33. Пайн (Pune J. B.), Linear programming on an analog computer. Trans. Paper AIEE, Princeton New-Jersey, Princ. Univ., pap. № 56 (1956), 147.

34. Петшиковский (Pietszykowski T.), An iteration method of linear programming. Zakład Aparatów Matematycznych PAN, Prace A, № 5, Warszawa, 1960.

35. Петшиковский (Pietszykowski T.), Application of the steepest ascent method to linear programming. Zakład Aparatów Matematycznych PAN, Prace A, № 11, Warszawa, 1961.

36. Робинсон Дж., Итеративный метод решения игр. Сб. «Матричные игры», Физматгиз, 1961, стр. 110—117.

37. Симоидс (Symonds G. H.), How to use linear programming. Petroleum Processing 10, № 5 (1955), 674—680.

38. Удзава Х., Итерационные методы вогнутого программирования. Сб. «Исследования по линейному и нелинейному программированию», 1962, стр. 228—245.

39. Фабнан (Fabian Tibor), A linear programming model of integrated iron and steel production. Management Sciend 4, № 4 (1958), 415—449.

40. Форд, Фулкерсон, Решение транспортной задачи. Сб. «Методы и алгоритмы решения транспортной задачи», Госстатиздат, 1963, стр. 61—72.

41. Хилти, Тэйлор, Жилеспай (Hilty D., Taylor R., Gillespie R.), Predicting minimum materials cost for stainless steel. Journ. of Metals 11, № 7 (1959), 458—464.

42. Чарнес, Купер, Меллон (Charnes A., Cooper W., Mellon B.), Blending aviation gasolines. Econometrica 20, № 2 (1952), 135—159.

43. Четыркин Е. М., Нормативные модели в экономике животноводства. Сб. статей под ред. Немчинова В. С. «Применение математики в экономических исследованиях», т. II, Соцэкгиз, 1962.

44. Шапиро Г. Н., Замечание о вычислительном методе в теории игр. Перев. с англ., Сб. «Матричные игры», Физматгиз, 1961, стр. 118—127.

45. Шка́, Сэв (Chueca A. et Séve A.), Nombre et débits optimaux des puits sur un gisement de pétrole. Revue de L'Institut Français du Pétrole, December, 15, № 12 (1960).

46. Эгервари (Egerváry E.), Matrixok combinatorins tulajdonsagairol. Math. Fiz. Lapok 38 (1931), 16—28.

47. Эрроу К. Дж., Гурвиц Л., Градиентный метод для вогнутого программирования, I—III. Перев. с англ., Сб. «Исследования по линейному и нелинейному программированию», ИЛ, 1962, стр. 175—188, 198—216.

48. Эрроу К. Дж., Соллоу Р. М., Градиентные методы отыскания условного максимума при ослабленных предположениях. Перев. с англ., сб. «Исследования по линейному и нелинейному программированию», ИЛ, 1962, стр. 246—264.

49. Юдин Д. Б., Гольштейн Е. Г., Об одном методе количественного анализа упрощенных экономических моделей. Сб. «Применение математики в экономических исследованиях», т. II, Соцэкгиз, 1962, стр. 136—199.

50. Юдин Д. Б., Гольштейн Е. Г., Линейное программирование (теория и конечные методы). Физматгиз, 1963.

51. Юдин Д. Б., Гольштейн Е. Г., Задачи и методы линейного программирования. Изд. 2-е, перераб. и дополн., изд-во «Советское радио», 1964.

Базис квазиплана 310
 — опорного плана 139, 196, 335, 350
 — системы векторов 91
 — сопряженный 273
 Базисные компоненты опорного плана 196, 351
 — составляющие псевдоплана 273
 — технологические способы производства 366, 367

Вершина множества 111
 Вектор возможностей производства 41
 — j -го технологического способа производства 41
 — запасов 41
 — затрат 41
 — n -мерный 83
 — невязок 311
 — ограничений задачи 10, 350
 — оценок факторов производства 158
 — разрешающий 176, 184
 — условий задачи 10, 350
 — расширенный 34

Векторов линейная комбинация 89

Векторы линейно зависимые 89
 — линейно независимые 89
 — ортогональные 87

Внебазисные переменные опорного плана 196, 351
 — — псевдоплана 273

Вспомогательная задача 378

Выпуклая оболочка множества 105

Выпуклое программирование 409

Геометрическая интерпретация задачи линейного программирования 32

Гиперплоскости линейно независимые 93

Гиперплоскость 92, 93
 — линейной формы задачи 143
 — опорная 100
 — разделяющая множества 96
 — — строго 96
 Грань q -мерная 111, 112

Длина вектора 88

Дробно-линейное программирование 60

Задача вспомогательная 308

— выбора 12
 — — рационального состава шихты 52
 — вырожденная 139
 — двойственная 149
 — невырожденная 139
 — об определении рациональных норм потребления продуктов питания 46
 — о графике работы сельскохозяйственных машин 64
 — о диете 46
 — о рациональном распределении посевных площадей 63, 64
 — о составлении экономного рациона откорма скота 63, 65
 — о специализации сельскохозяйственного производства 64
 — переработки нефтепродуктов 62
 — планирования кормовой базы 69
 — прямая 149
 — расширенная 300, 308
 — сопряженная 149
 — — расширенная 300
 — со смешанными условиями 9

- Итерация 208
 — большая 380
 — малая 380
- Каноническая форма 24
- Квадратичное программирование 409
- Квазиплани 310
- Компоненты вектора 83
 — плана 10
- Конус выпуклый 96
 — многогранный 123, 125
- Критерий оптимальности плана задачи 178
- Линейная форма задачи 9
- Линейное программирование 8
- Луч 94
- Математическое программирование 5
- Матрица условий задачи 9
- Метод двусторонних оценок 328
 — последовательного сокращения невязок 307, 372
 — — улучшения плана 194, 371
 — — уточнения оценок 271, 372
 — симплексный 194
- М-задача 265
- Минор 90
- Многогранник выпуклый 108, 123
 — условий задачи 109
- Многообразие линейное 92, 93
- Множество выпуклое многогранное 108, 122
 — замкнутое 94
 — многогранное условий задачи 109, 126
 — ограниченное 95
- Множители Лагранжа 186
 — разрешающие 176
- Мультипликативная форма второго алгоритма 251
- Направляющий вектор гиперплоскости 93
 — — прямой 94
- Невязка 305
 — квазиплана 311
- Неравенство Буяковского 87
- Норма вектора 88
- Нулевой вектор 86
- Область определения задачи 9
 — — линейной формы задачи 126
- Ограничение жесткое 109
 — нежесткое 109
- Окрестность точки A 94
- Опорная прямая 30
- Опорный план сопряженной задачи 273
- Острие конуса 124
- Отрезок 94
- Оценки векторов условий относительно данного базиса 203
 — условий задачи относительно данного базиса 204
 — факторов предварительные 159
- Пара задач взаимосопряженная 156
 — — двойственная 156
- Параметр элементарного преобразования 357
- Пары двойственных условий 171
- План задачи 10
 — опорный 128, 138, 196, 334, 350
 — — вырожденный 139
 — — невырожденный 139, 141, 335
 — оптимальный 10
 — цен 365
- Подматрица 90
- Подпространство 91
- Полупространство 94
 — верхнее, нижнее 143
- Полупрямая 94
- Преобразование элементарное 202
- Признак оптимальности квазиплана 311
 — — опорного плана 352
 — — — —, вторая форма 200
 — — — —, первая форма 199
 — — псевдоплана 275
- Произведение вектора на число 86
- Пространство евклидово 88
 — n -мерное векторное 392
- Процесс решения игры 392
- Прямая 92
- Псевдоплан 273

Размериость выпуклого множества 103
 — многогранного множества 110
 — пространства 90
 Разность векторов 86
 Разрешимая задача 11, 129
 Ранг матрицы 91
 — системы векторов 91
 — — линейных ограничений 110
 Ребро многогранного множества 112
 Решение задачи 10
 — игры 392
 — — в смешанных стратегиях 392

Система ограничений линейно независимая 110
 — уравнений регулярная в точке X 185
 — условий задачи 9
 — — множества 108
 Скалярное произведение векторов 87
 Сопряженная задача невырожденная 273
 — каноническая форма 26
 Составляющие вектора 83
 — плана 10
 Стратегия смешанная 391
 — чистая 391
 Сумма векторов 83
 — множеств 118
 Сходимость последовательности векторов 89

Точка множества внешняя 94
 — — внутренняя 94
 — — граничная 94

Точка множества крайняя 104
 — седловая 189
 Теорема Больцаио — Вейерштрасса 95
 — двойственности вторая 171
 — — первая 163
 — об опорной гиперплоскости 101
 — о представлении многогранного множества 118
 — о разделяющей гиперплоскости 97
 — о разрешающих векторах 183
 — о разрешимости задачи линейного программирования 134
 — о существовании опорного плана 129
 — — — — решения 131

Уравнение гиперплоскости 93
 Условие закрепленное 171
 — свободное 171
 Условия однородные 25

Факторы производственные 37
 Фиктивные технологические процессы производства 364
 Функция Лагранжа 186
 — платежная 392

Целочисленное программирование 51
 Цена игры 392

Шар 94

Элементарное преобразование в методе последовательного сокращения невязок 312
 — — — — улучшения плана 202
 — — — — уточнения оценок 278

Д71100

А.БЛОДИН
Е.ГОДЪШТЕЙН

**Линейное
программирование**
теория, методы
и приложения

Линейное программирование