Инновационный курс

«Дополнительные главы исследования операций» (64 часа)

Аннотация

Обязательный курс для магистров 2-го года, обучающихся по программе "Математическое и информационное обеспечение экономической деятельности".

Читается в 4-м семестре.

Лекции 32 часа.

Самостоятельная работа 32 часа.

Зачет и экзамен в 4-м семестре.

За курс отвечает кафедра исследования операций.

Авторы программы: профессор Васин А.А., доцент Морозов В.В.

Лектор 2005/06 уч. года: доцент Морозов В.В.

Содержание курса

Введение. Исследование операций в экономике. Принятие решений в условиях отсутствия неконтролируемых факторов. Непрерывные и дискретные задачи оптимизации.

Задача линейного программирования. Постановка задачи линейного ЛП и ее эквивалентные формы: каноническая, стандартная и общая. Задачи планирования производства и оптимизации структуры страхового портфеля. Базисные и реберные решения. Вырожденные и невырожденные базисные решения. Понятие базиса. Алгебраическая характеризация базисных допустимых решений задачи ЛП. Вершины и ребра многогранника ограничений.

Симплекс-метод. Движение по ребру от одного базисного решения к другому. Эквивалентное преобразование задачи ЛП. Операция замещения. Симплекс-таблица как инструмент симплекс-метода. Пример зацикливания симплекс-метода. Преодоление зацикливания. Правило Блэнда. Методы поиска начального БДР: метод искусственного базиса, метод замещений.

Теория двойственности. Двойственная задача ЛП и метод ее выписывания. Интерпретация двойственных переменных как оценок дефицитности ресурсов. Теорема двойственности. Свойство дополняющей нежесткости.

Методы, использующие двойственность. Двойственный симплекс-метод. Симплекс-метод для задачи ЛП с двусторонними ограничениями. Задача о формировании оптимального портфеля ценных бумаг.

Транспортная задача. Общий метод поиск начального базисного решения транспортной задачи. Методы северо-западного угла и минимального элемента. Свойства базисного решения.Метод потенциалов для поиска оптимального решения транспортной задачи.

Метод ветвей и границ. Общая схема метода ветвей и границ. Задача о назначении. Задача целочисленного ЛП и ее решение методом ветвей и границ. Задача о рюкзаке. Метод Балаша для задачи булева программирования. Задача об оптимальном инвестировании.

Динамическое программирование. Метод динамического программирования для задачи с сепарабельной целевой функцией. Задача о размещении рекламных объявлений. Динамическое программирование для задачи с мультипликативной функцией. Задача оптимизации надежности технического устройства.)

Распределение часов курса по темам и видам работ

№ п/п	Наименование разделов и тем	Всего (часов)	Лекции (часов)	Самостоятельная работа
1	Задача линейного программирования	6	3	3
2	Симплекс-метод	16	8	8
3	Теория двойственности	8	4	4
4	Методы, использующие двойственность	12	6	6
5	Транспортная задача	6	3	3
6	Метод ветвей и границ	12	6	6
7	Динамическое программирование	4	2	2
	ИТОГО	64	32	32

Список литературы

[1] Ашманов С.А. Линейное программирование. М.: Наука, 1981.

[2] Гольштейн Е.Г., Юдин Д.Б. Новые направления в линейном программировании. М.: Советское радио, 1966.

[3] Гольштейн Е.Г., Юдин Д.Б. Задача линейного программирования транспортного типа. М.: Наука, 1969.

[4] Данциг Д.Б. Линейное программирование, его применения и обобщения. М.: Прогресс, 1966.

[5] Зуховицкий С.И., Радчик И.А. Математические методы сетевого планирования. М.: Наука, 1965.

[6] Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.

[7] Юдин Д.Б., Гольштейн Е.Г. Линейное программирование. М.: Наука, 1963.

[8] Ху Т. Целочисленное программирование и потоки в сетях. М.: Мир, 1974.

.

Список вопросов, выносимый на экзамен

1. Алгебраическая характеризация БДР задачи ЛП.
2. Движение по ребру от одного базисного решения к другому.
3. Преобразование задачи ЛП и симплекс-метод.
4. Преодоление зацикливания.
5. Методы поиска начального БДР.
6. Двойственная задача ЛП, ее интерпретация и метод выписывания.

7. Теорема двойственности и ее следствия.
8. Двойственный симплекс-метод.
9. Симплекс-метод для задачи ЛП с двусторонними ограничениями.
10. Поиск начального базисного решения транспортной задачи и его свойства.
11. Метод поиска оптимального решения транспортной задачи.
12. Задача целочисленного ЛП и ее решение методом ветвей и границ.
13. Метод Балаша для задачи булевого программирования.
14. Метод динамического программирования для задачи с сепарабельной целевой функцией.
15. Метод динамического программирования для задачи с мультипликативной целевой функцией.

Конспект лекций
«Дополнительные главы исследования операций»

§1. Постановка задачи ЛП и ее эквивалентные формы

Определим задачу линейного программирования (ЛП) в *стандартной форме.* Требуется максимизировать линейную *целевую* функцию при линейных ограничениях:

$$\begin{gathered} Z = \sum_{j=1}^{n} c_j x_j \ \to \ max \\ \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, ..., m, \qquad x_j \geqslant 0, \ j = 1, ..., n. \end{gathered} \tag{1.1}$$

Задачу (1.1) можно интерпретировать в терминах планирования производства. Пусть

b_i – количество ресурса i-го типа, используемого в процессе производства n видов продукции;

$A = (a_{ij})_{m \times n}$ – технологическая матрица, где a_{ij} – количество ресурса i-го типа, необходимого для производства единицы продукции j-го вида;

c_j – стоимость единицы продукции j-го вида.

План производства задается величинами x_j, $j = 1, ..., n$, – количествами выпускаемой продукции каждого вида. Задача (1.1) состоит в нахождении плана производства, максимизирующего стоимость выпущенной продукции при заданных ограничениях на ресурсы.

Пример 1.1. Оптимизация структуры страхового портфеля.

Медицинская страховая компания производит страхование по двум видам заболеваний, требующим сложных хирургических операций. Вероятность первого заболевания равна 0.0048, а продолжительность пребывания в больнице составляет в среднем 30 дней. Аналогичные цифры по второму заболеванию равны 0.0018 и 60. Компания может обеспечить в год 240 и 144 операций по первому и второму заболеванию, а страховые премии (взносы) составляют 2 и 3 условных единицы соответственно. Компания имеет в своем распоряжении 32 больничных номера. Требуется определить оптимальное содержимое страхового портфеля, обеспечивающее максимальную страховую премию при заданных ограничениях. Сформулируем задачу ЛП, основываясь на средних величинах.

Пусть x_j – число десятков тысяч договоров, заключаемых по j-му виду страхования, $j = 1, 2$. По первому виду страхования ожидаемое число страховых случаев $10^4 x_1 \cdot 0.0048 =$

$48x_1$ не должно превосходить 240. Поэтому получаем неравенство $x_1 \leqslant 5$. Аналогичное рассуждение для второго вида страхования приводит к неравенству $x_2 \leqslant 8$. Поскольку в течении года в одном больничном номере можно разместить 12 пациентов с первым видом заболевания или 6 пациентов со вторым, то имеем еще одно ограничение

$$\frac{48x_1}{12} + \frac{18x_2}{6} \leqslant 32.$$

Получаем задачу ЛП

$$Z = 2x_1 + 3x_2 \ \rightarrow \ max$$

$$4x_1 + 3x_2 \leqslant 32, \qquad 0 \leqslant x_1 \leqslant 5, \qquad 0 \leqslant x_2 \leqslant 8.$$

Определение. Любой вектор $x = (x_1, ..., x_n)$ будем называть *решением*[1] задачи (1.1). Значение целевой функции $Z(x)$ на решении x будем называть *стоимостью* этого решения по аналогии с задачей планирования производства. Решение называется *допустимым*, если оно удовлетворяет ограничениям задачи (1.1). Множество допустимых решений задачи (1.1) обозначим через X. Допустимое решение x^* называется *оптимальным*, если оно реализует максимум целевой функции на множестве всех допустимых решений, т.е. $Z(x^*) = \max\limits_{x \in X} Z(x)$.

Заметим, что нулевое решение $\mathbf{0} = (0, ..., 0) \in E^n$ допустимо, если все коэффициенты b_i, $i = 1, ..., n$, неотрицательны.

Задача ЛП может быть задана в форме, отличной от стандартной. Если основные ограничения задачи − равенства, а условия неотрицательности переменных сохраняются, то говорят, что задача ЛП имеет *каноническую форму*:

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

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

Наконец, говорят, что задача ЛП имеет *общую форму*, если в ней присутствуют *свободные* переменные, на которые не наложено условие неотрицательности, при этом часть ограничений задана в виде равенств, а другая часть − в виде неравенств:

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

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \ i = 1, ..., k < m,$$

$$\sum_{j=1}^{n} a_{ij} x_j \ = b_i, \ i = k+1, ..., m, \quad x_j \geqslant 0, \ j = 1, ..., s < n.$$

[1] Другая терминология, используемая для обозначения вектора x : план, точка. Рассматривая задачу ЛП как модель операции, можно решение x считать стратегией, а целевую функцию Z − критерием эффективности оперирующей стороны.

Удобно говорить об *основных* ограничениях, в которых используются коэффициенты a_{ij}. Отметим, что выделение свободных переменных может быть весьма условным. Например, если ограничение $x_1 \geqslant 0$ перевести в основное, расширив матрицу A, то переменная x_1 станет свободной.

Покажем, что все введенные формы задачи ЛП эквивалентны. Действительно, каноническую форму легко свести к стандартной. Для этого достаточно каждое ограничение-равенство

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

записать в виде двух неравенств

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

Покажем, что задача ЛП в общей форме также сводится к стандартной. Пусть x_j – свободная переменная. Тогда ее представляют в виде разности неотрицательных переменных: $x_j = x'_j - x''_j$, где $x'_j \geqslant 0,\ x''_j \geqslant 0$. Наконец, каждое ограничение-равенство, как было показано, можно заменить двумя неравенствами.

Запишем теперь в канонической форме задачу ЛП, заданную первоначально в стандартной форме (1.1). Для этого ограничения-неравенства превратим в равенства, вводя *слабые*[1] переменные $x_{n+i},\ i = 1, ..., m$. В результате получаем задачу ЛП

$$\begin{gathered} Z = \sum_{j=1}^{n} c_j x_j \ \rightarrow \ max \\ \sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i,\ i = 1, ..., m, \\ x_j \geqslant 0,\ j = 1, ..., n+m. \end{gathered} \tag{1.2}$$

Будем говорить, что задача (1.2) имеет *диагональную* форму относительно переменных $x_{n+1}, ..., x_{n+m}$. Соответствующие этим переменным столбцы матрицы ограничений образуют единичную матрицу, а целевая функция от них не зависит. Расширенная матрица системы уравнений, входящих в ограничения задачи (1.2), имеет вид $(AI_m|b)$, где I_m – единичная матрица размерности m. Обозначим через A_j j-й столбец матрицы (AI_m). В частности, $A_{n+i} = e_i \in E^m$ – вектор, в котором все компоненты равны нулю, за исключением i-й, равной единице.

Отметим, что диагональная форма может быть записана относительно другого набора из m переменных. Поэтому в общем случае задачу ЛП можно привести к диагональной форме несколькими возможными способами.

Заметим также, что при приведении общей формы задачи к канонической, можно исключить все свободные переменные x_j вместо введения неотрицательных переменных x'_j и x''_j. Подробно этот вопрос рассматривается в § 6.

Между допустимыми решениями задач (1.1) и (1.2) имеется взаимно однозначное соответствие. В самом деле, предположим, что $x = (x_1, ..., x_n)$ – допустимое решение задачи

[1] Иногда их называют переменными недостатка.

(1.1). Тогда величины

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \geqslant 0, \quad i = 1, ..., m,$$

определяются однозначно и $\overline{x} = (x_1, ..., x_{n+m})$ – допустимое решение задачи (1.2). Обратное утверждение очевидно.

Отметим, что любую задачу ЛП, заданную в канонической форме, можно свести к диагональной форме, используя преобразования Гаусса-Жордана над системой линейных уравнений, а также исключая соответствующие единичной матрице переменные из целевой функции.

Пример 1.2. Пусть задана задача ЛП в канонической форме

$$Z = 3x_1 - 2x_2 + x_3 \to max$$

$$x_1 + x_2 + x_3 = 5, \; 3x_1 + 2x_2 + x_3 = 3,$$

$$x_j \geqslant 0, \; j = 1, 2, 3.$$

Преобразуем ее к диагональной форме (1.2) относительно переменных x_2, x_3. Для этого вычтем из второго уравнения первое, а затем полученное уравнение $2x_1 + x_2 = -2$ вычтем из первого. В результате получим уравнение $-x_1 + x_3 = 7$. Исключим переменные x_2 и x_3 из целевой функции. В итоге задача приобретает искомую форму

$$Z = 8x_1 + 11 \to max$$

$$2x_1 + x_2 = -2, \;\; -x_1 + x_3 = 7, \;\; x_j \geqslant 0, \; j = 1, 2, 3.$$

Упражнения и задачи.

1.1. Преобразовать к стандартной и канонической формам задачу ЛП

$$Z = 2x_1 - x_2 - 5x_3 \to max$$

$$4x_1 + x_2 + x_3 = 2, \; -x_1 + 2x_2 + x_3 \geqslant 2, \;\; -x_1 + 2x_2 + x_3 \leqslant 4,$$

$$x_1 \geqslant 0, \; x_3 \geqslant 0.$$

1.2. Используя преобразования Гаусса-Жордана, запишите в диагональной форме относительно переменных x_1, x_2 задачу ЛП, заданную в канонической форме

$$Z = 4x_1 + 2x_2 - x_3 + x_4 \;\to\; max$$

$$x_1 - 2x_2 + x_3 + x_4 = 1, \; -x_1 + x_2 - 3x_3 + 2x_4 = 2, \quad x_j \geqslant 0, \; j = 1, 2, 3, 4.$$

§2. Базисные и реберные решения

Рассмотрим задачу ЛП в произвольной канонической форме

$$Z = \sum_{j=1}^{n} c_j x_j \ \rightarrow \ max$$
$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, ..., m, \qquad x_j \geqslant 0, \ j = 1, ..., n. \tag{2.1}$$

Для решения $x = (x_1, ..., x_n)$ этой задачи положим $J(x) = \{j \mid x_j \neq 0\}$. Если x – допустимое решение задачи (2.1), то компоненты $x_j, \ j \in J(x)$, положительны, а остальные равны нулю.

Определение. Решение $x^0 = (x_1^0, ..., x_n^0)$, удовлетворяющее ограничениям-равенствам задачи (2.1), называется *базисным*, если система столбцов $\{A_j \mid j \in J(x^0)\}$ матрицы A линейно независима.

Отметим, что в случае $b_i = 0, \ i = 1, ..., m$, нулевое решение $\mathbf{0} \in E^n$ с $J(\mathbf{0}) = \emptyset$ также будем считать базисным.

Нетрудно видеть, что ненулевое базисное решение x^0 – единственное решение системы уравнений

$$\sum_{j \in J(x^0)} x_j A_j = b, \quad x_j = 0, \ j \notin J(x^0). \tag{2.2}$$

Поскольку существует конечное число линейно независимых систем, составленных из столбцов матрицы A, то число базисных решений конечно.

Для базисных допустимых решений будем использовать аббревиатуру БДР. Пусть X – множество всех допустимых решений задачи (1.1). Множество X представляет собой *многогранное множество* в E^n. Ограниченное множество X называется *многогранником.*

Определение. Решение $x^0 \in X$ называется *крайней точкой (вершиной)* множества X, если не существует таких различных решений $x', x'' \in X$ и такого числа $\lambda \in (0, 1)$, что решение x^0 представимо в виде $x^0 = \lambda x' + (1 - \lambda) x''$.

Иначе говоря, крайняя точка множества X не может быть внутренней точкой отрезка, соединяющего два допустимых решения.

Теорема 2.1. Для того чтобы решение $x^0 \in X$ было БДР, необходимо и достаточно, чтобы оно являлось крайней точкой множества X.

Определение. *Базисом* называется любая система $\mathcal{B} = \{A_j \mid j \in J\}$, состоящая из m линейно независимых столбцов матрицы A. При этом J будем называть *множеством базисных номеров* или сокращенно МБН. Ясно, что $\mathcal{B}$ является базисом пространства E^m в обычном алгебраическом смысле. В частности, любой вектор из E^m единственным образом разлагается по базисным столбцам.

Пусть x^0 – базисное решение. Заметим, что число элементов $|J(x^0)|$ множества $J(x^0)$ не превосходит m, поскольку система столбцов $\{A_j \mid j \in J(x^0)\}$ линейно независима. Если $|J(x^0)| = m$, то решение x^0 называется *невырожденным*, а в противном случае (при $|J(x^0)| < m$) – *вырожденным.* Невырожденному базисному решению x^0 соответствует единственный базис $\mathcal{B} = \{A_j \mid j \in J(x^0)\}$. Для вырожденного базисного решения линейно независимую систему столбцов $\{A_j \mid j \in J(x^0)\}$ можно дополнить до базиса $\mathcal{B} = \{A_j \mid j \in J\}$, $J(x^0) \subset J$. Вырожденным базисным решениям обычно соответствуют несколько базисов.

Если задача ЛП задана в диагональной форме (1.2), то (AI_m) – матрица системы уравнений, образующих основные ограничения. Отметим, что в этом случае решение $x^{(1)} = (\underbrace{0, ..., 0}_{n \text{ раз}}, b_1, ..., b_m)$ всегда является базисным с МБН $J^{(1)} = \{n+1, ..., n+m\}$.

Пример 2.1. Рассмотрим систему неравенств

$$x_1 + x_2 \leqslant 1, \;\; x_1 + x_3 \leqslant 1, \;\; x_2 + x_3 \leqslant 1, \;\; x_1, x_2, x_3 \geqslant 0.$$

Перейдем к соответствующей системе равенств

$$\begin{aligned} x_1 + x_2 \quad + x_4 \quad\quad &= 1, \\ x_1 + \quad x_3 + \quad x_5 \quad &= 1, \\ x_2 + x_3 + \quad\quad x_6 &= 1, \\ x_j \geqslant 0, \; j = 1, ..., 6. \end{aligned}$$

Здесь

$$(AI_3) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Вырожденному БДР $x^1 = (1, 0, 0, 0, 0, 1)$ можно сопоставить четыре базиса:

$$\mathcal{B}^1 = \{A_1, A_2, A_6\}, \; \mathcal{B}^2 = \{A_1, A_3, A_6\}, \; \mathcal{B}^3 = \{A_1, A_4, A_6\},$$

$$\mathcal{B}^4 = \{A_1, A_5, A_6\}.$$

Заметим, что $x' = (1, 1, 0, -1, 0, 0)$ – базисное, но не допустимое решение.

Пример 2.2. Пусть ограничения задачи (1.1) имеют следующий вид:

$$x_1 \leqslant 1, \quad 0 \cdot x_1 \leqslant 0, \quad x_1 \geqslant 0.$$

Здесь

$$(AI_2) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Тогда $x^0 = (1, 0, 0)$ – вырожденное БДР с единственным базисом $\mathcal{B} = \{A_1, A_3\}$.

Определение. Решение x^0, удовлетворяющее ограничениям-равенствам задачи (1.2), называется *реберным* , если либо $|J(x^0)| = r$ (случай базисного решения), либо $|J(x^0)| = r+1$, где r – ранг системы столбцов $\{A_j \mid j \in J(x^0)\}$.

Пример 2.3. Для ограничений

$$x_1 + x_3 = 1, \; x_2 + x_4 = 1, \; x_j \geqslant 0, \; j = 1, 2, 3, 4,$$

точки $(1, x_2, 0, x_4)$, где $x_2 + x_4 = 1,\ x_2, x_4 \geqslant 0$, являются примерами реберных решений.

Введем теперь понятие *ребра* множества X. Пусть x^0 – допустимое реберное, но не базисное решение. Тогда $|J(x^0)| = r + 1$ и решение x^0 удовлетворяет системе уравнений (2.2). Возьмем x' – ненулевое решение соответствующей однородной системы

$$\sum_{j \in J(x^0)} x_j A_j = \mathbf{0}, \quad x_j = 0,\ j \notin J(x^0).$$

Решение $x(t) = x^0 + tx'$, где t – параметр, также удовлетворяет неоднородной системе (2.2). Для того чтобы $x(t) \in X$, необходимо потребовать выполнение неравенств $x_j(t) \geqslant 0,\ j \in J(x^0)$. Отметим, что при достаточно близком к нулю t последние неравенства выполнены как строгие.

Совокупность построенных допустимых реберных решений $x(t)$ назовем ребром множества X. Неравенства $x_j(t) \geqslant 0,\ j \in J(x^0)$, задают следующие возможные ограничения на параметр t :

1) t принадлежит отрезку. Тогда ребро – отрезок в E^n.

2) t принадлежит полупрямой. Тогда ребро – луч в E^n.

Лемма 2.1. Пусть решение $x(t^0)$ – крайняя точка ребра. Тогда оно является БДР.

§3. Симплекс-метод. Движение по ребру

Симплекс-метод применяется для поиска оптимального решения задачи ЛП. Он также позволяет определить, совместны ли ограничения задачи и ограничена ли сверху целевая функция на множестве допустимых решений.

Рассмотрим задачу ЛП в диагональной форме (1.2). Обозначим через $\overline{X}$ множество допустимых решений $\overline{x} = (x_1, ..., x_{n+m})$ задачи (1.2). Основным шагом (итерацией) симплекс-метода является перемещение по ребру множества $\overline{X}$ от одного БДР к другому. При этом целевая функция возрастает.

Пусть $b_i \geqslant 0,\ i = 1, ..., m$. Тогда возьмем $x^{(1)} = (\underbrace{0, ..., 0}_{n \text{ раз}}, b_1, ..., b_m)$ в качестве начального БДР с МБН $J^{(1)} = \{n+1, ..., n+m\}$. Поиск *начального БДР* для произвольных b_i обсуждается в § 6.

Построим ребро, выходящее из $x^{(1)}$ и принадлежащее множеству $\overline{X}$. Для этого возьмем вне МБН $J^{(1)}$ какой-нибудь номер p (в данном случае $p \leqslant n$). Будем увеличивать от нуля переменную x_p, а остальные небазисные переменные оставим нулевыми. Для того чтобы решение удовлетворяло ограничениям-равенствам задачи (1.2), необходимо изменить базисные компоненты. В результате получаем решение

$$\tilde{x} = (0, ..., 0, x_p, 0, ..., 0, b_1 - a_{1p}x_p, ..., b_m - a_{mp}x_p).$$

Решение $\tilde{x}$ допустимо, если $b_i - a_{ip}x_p \geqslant 0,\ i = 1, ..., m$, поскольку в этом случае все его компоненты будут неотрицательными. Если окажется, что $a_{ip} \leqslant 0,\ i = 1, ..., m$, то при $x_p \to +\infty$ решение $\tilde{x}$ будет оставаться допустимым. Это означает, что из точки $x^{(1)}$ выходит ребро-луч с *направляющим* вектором

$$(0, ..., 0, \underset{p}{1}, 0, ..., 0, -a_{1p}, ..., -a_{mp}).$$

Предположим, что при некотором i выполнено $a_{ip} > 0$. Тогда получим ограничение на переменную x_p сверху:

$$0 \leqslant x_p \leqslant \theta \stackrel{def}{=} \min_{i:\, a_{ip}>0} \frac{b_i}{a_{ip}} = \frac{b_l}{a_{lp}}.$$

При $0 \leqslant x_p \leqslant \theta$ решение $\tilde{x}$ будет допустимым и пробегает ребро-отрезок, если $\theta > 0$. Одним концом этого отрезка будет решение $x^{(1)}$, а другим $-$ решение

$$x^{(2)} = (0, ..., 0, \underset{p}{\theta}, 0, ..., 0, b_1 - a_{1p}\theta, ..., \underset{n+l}{0}, ..., b_m - a_{mp}\theta).$$

Из леммы 2.1 вытекает, что решение $x^{(2)}$ является базисным с новым МБН $J^{(2)} = (J^{(1)} \setminus \{n+l\}) \cup \{p\}$. Таким образом, построенному БДР $x^{(2)}$ мы сопоставили новый базис с МБН $J^{(2)}$.

В результате перемещения в точку $x^{(2)}$ переменная x_{n+l} стала небазисной, а переменная x_p $-$ базисной. Как говорят, произошла *операция замещения:* Как говорят, произошла операция *замещения:* столбец A_{n+l} и соответствующая переменная x_{n+l} были выведены из базиса, а столбец A_p и переменная x_p введены в него. При этом l-я строка и p-й столбец, а также элемент a_{lp}, стоящий на их пересечении, называются *ведущими*[1].

На рассматриваемом ребре целевая функция записывается в виде $Z(\tilde{x}) = c_p x_p$, поскольку она не зависит от базисных переменных, соответствующих решению $x^{(1)}$. Если $c_p > 0$ и $\theta > 0$, то целевая функция вдоль ребра возрастает. Поэтому номер ведущего столбца p выбирают из условия $c_p > 0$.

Пример 3.1. Запишем задачу ЛП из примера 1.1 в диагональной форме:

$$Z = 2x_1 + 3x_2 \ \rightarrow \ max$$

$$x_1 + x_3 = 5, \ \ x_2 + x_4 = 8, \ \ 4x_1 + 3x_2 + x_5 = 32, \ \ x_j \geqslant 0, \ j = 1, ..., 5.$$

Матрица ограничений имеет вид

$$(AI_3|b) = \left(\begin{array}{ccccc|c} 1 & 0 & 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 & 0 & 8 \\ 4 & 3 & 0 & 0 & 1 & 32 \end{array}\right)$$

Начальному базисному решению $x^{(1)} = (0, 0, 5, 8, 32)$ соответствует МБН $J^{(1)} = \{3, 4, 5\}$. Будем увеличивать вторую компоненту $x_2 = t$. В результате получим реберное решение $\tilde{x} = (0, t, 5, 8 - t, 32 - 3t)$. Оно является допустимым, если выполняются неравенства $t \geqslant 0, \ 8 - t \geqslant 0, 32 - 3t \geqslant 0$. Отсюда следует, что $0 \leqslant t \leqslant \theta = \min[8, 32/3] = 8$. При $t = 8$ получаем новое базисное решение $x^{(2)} = (0, 8, 5, 0, 8)$ с МБН $J^{(2)} = \{2, 3, 5\}$.

Отметим случаи, когда алгоритм останавливает свою работу.

1) $a_{ip} \leqslant 0, \ i = 1, ..., m, \ \ c_p > 0$.

Тогда из точки $x^{(1)}$ выходит луч, принадлежащий $\overline{X}$, вдоль которого целевая функция неограниченно возрастает. Следовательно, оптимального решения не существует.

2) $c_j \leqslant 0, \ \ j = 1, ..., n$.

Тогда можно утверждать, что решение $x^{(1)}$ является оптимальным. Действительно, для всякого $\overline{x} \in \overline{X}$

$$Z(\overline{x}) = \sum_{j=1}^{n} c_j x_j \leqslant 0 = Z(x^{(1)}).$$

[1]Элемент a_{lp} называют также *разрешающим.*

Алгоритм продолжает работу, если $c_p > 0$ и существует такой номер i, что $a_{ip} > 0$. Тогда переходим к БДР $x^{(2)}$. Отметим, что в случае $b_l = 0$ $x^{(1)}$ – вырожденное БДР, величина θ равна нулю, а решения $x^{(1)}$ и $x^{(2)}$ совпадают. Здесь реального перемещения вдоль ребра не происходит, хотя базис меняется.

§4. Симплекс-метод. Эквивалентные преобразования

Чтобы продолжить работу симплекс-метода и перейти к следующему БДР $x^{(3)}$, необходимо эквивалентно преобразовать задачу (1.2). Отметим свойство, которому удовлетворяет начальное БДР $x^{(1)}$ по отношению к задаче (1.2).

Свойство (α) :

а) базисные столбцы задачи (1.2), соответствующие решению $x^{(1)}$, образуют единичную матрицу;

б) целевая функция Z зависит только от небазисных переменных.

Условие а) позволило явно определить реберное решение $\tilde{x} \in [x^{(1)}, x^{(2)}]$, а условие б) – записать целевую функцию на отрезке $[x^{(1)}, x^{(2)}]$ в виде $Z(\tilde{x}) = c_p x_p$, а также сформулировать условие оптимальности. Поэтому для продолжения вычислений нам необходимо так эквивалентно преобразовать задачу (1.2), чтобы свойство (α) было выполнено для БДР $x^{(2)}$.

Какие преобразования задачи (1.2) возможны? Поскольку основные ограничения задачи (1.2) заданы в форме линейных уравнений, то их можно подвергнуть элементарным преобразованиям Гаусса-Жордана: умножать уравнение на число, прибавлять к одному уравнению другое, умноженное на число.

Пусть a_{lp} – ведущий элемент. Для выполнения условия а) свойства (α) необходимо элементарными преобразованиями со строками матрицы $(AI_m|b)$ добиться, чтобы p-й базисный столбец стал равен e_l. В результате базисные столбцы, соответствующие решению $x^{(2)}$, будут образовывать единичную матрицу.

Еще одно преобразование можно сделать с целевой функцией. Запишем ее в следующем виде:

$$Z = \sum_{j=1}^{n} c_j x_j - \frac{c_p}{a_{lp}} \Big(\sum_{j=1}^{n} a_{lj} x_j + x_{n+l} - b_l \Big). \tag{4.1}$$

Заметим, что если решение $\overline{x}$ удовлетворяет ограничениям-равенствам задачи (1.2), то на нем вычитаемое выражение обращается в нуль и фактически целевая функция не меняется. После преобразования (4.1) функция Z не зависит от базисных переменных решения $x^{(2)}$, поскольку коэффициент при переменной x_p равен нулю. Тем самым, выполнено условие б) свойства (α).

Запишем формулу (4.1) в виде

$$Z = \sum_{j=1}^{n+m} c'_j x_j + c'_0, \tag{4.2}$$

где $c'_0 = c_p b_l / a_{lp} = c_p \theta$. Покажем, что $c'_0 = Z(x^{(2)})$. Действительно, как выше отмечалось, в (4.1) целевая функция Z не зависит от базисных переменных, соответствующих решению $x^{(2)}$. Поэтому при подстановке решения $x^{(2)}$ в формулу (4.2) сумма в ее правой части обратится в нуль. Теперь перейдем от решения $x^{(2)}$ к решению $x^{(3)}$ тем же способом, каким мы переходили от решения $x^{(1)}$ к решению $x^{(2)}$.

Все проделанные преобразования задачи (1.2) можно свести к преобразованию так называемой *симплекс-таблицы*. *Начальная* симплекс-таблица имеет вид

Z	$-c_1$	$\dots$	$-c_n$	0	$\dots$	0	0
x_{n+1}	a_{11}	$\dots$	a_{1n}	1	$\dots$	0	b_1
$\vdots$		$\vdots$			$\ddots$		$\vdots$
x_{n+m}	a_{m1}	$\dots$	a_{mn}	0	$\dots$	1	b_m .

Нулевой столбец[1] таблицы содержит переменную Z и базисные переменные $x_{n+1}, ..., x_{n+m}$, соответствующие базисному решению $x^{(1)}$. Последний столбец таблицы является значением первого столбца. А именно, в правом верхнем углу располагается 0 – значение целевой функции на решении $x^{(1)}$, а ниже – базисные компоненты $b_1, ..., b_m$ решения $x^{(1)}$. Нулевая строка таблицы содержит коэффициенты целевой функции, взятые со знаком минус[2]. Она соответствует уравнению $Z - \sum\limits_{j=1}^{n} c_j x_j = 0$. Чтобы перейти от решения $x^{(1)}$ к решению $x^{(2)}$, необходимо выбрать небазисную компоненту x_p таким образом, чтобы элемент нулевой строки p-го столбца был отрицательным ($-c_p < 0$). При этом p-й столбец будет ведущим.

Выбор ведущего столбца может быть неоднозначным, если нулевая строка содержит несколько отрицательных элементов. Затем, как уже указывалось, находится величина

$$\theta = \min_{i:\ a_{ip}>0} \frac{b_i}{a_{ip}} = \frac{b_l}{a_{lp}}.$$

Переменная x_{n+l} выводится из базиса. Поскольку строки симплекс-таблицы соответствуют линейным уравнениям, над ними можно осуществлять преобразования Гаусса-Жордана. Например, преобразованию целевой функции (4.1) соответствует прибавление к нулевой строке симплекс-таблицы l-й строки, умноженной на c_p/a_{lp}. В результате на пересечении нулевой строки и p-го столбца появится нулевой элемент. Преобразуем таблицу так, чтобы ее новый базисный p-й столбец был равен $e_l \in E^{m+1}$. В нулевом столбце таблицы переменную x_{n+l} следует заменить на x_p. Преобразованная симплекс-таблица имеет вид

Z	$\cdots$	$-c_j + c_p a_{lj}/a_{lp}$	$\cdots$	0	$\cdots$	$c_p\theta$
x_{n+1}	$\cdots$	$a_{1j} - a_{1p}a_{lj}/a_{lp}$	$\cdots$	0	$\cdots$	$b_1 - a_{1p}\theta$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_p	$\cdots$	a_{lj}/a_{lp}	$\cdots$	1	$\cdots$	θ
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n+m}	$\cdots$	$a_{mj} - a_{mp}a_{lj}/a_{lp}$	$\cdots$	0	$\cdots$	$b_m - a_{mp}\theta$.

Заметим, что в последнем столбце таблицы появились базисные компоненты решения $x^{(2)}$ и значение целевой функции $Z(x^{(2)}) = c_p\theta$. Совокупность произведенных преобразований таблицы будем называть *операцией замещения*.

[1] Нулевой столбец (и ниже нулевая строка) в данном контексте – это столбец (строка) симплекс-таблицы с нулевым номером.

[2] Отметим, что в ряде руководств эти коэффициенты помещаются в последней строке таблицы, а коэффициенты b_i – в нулевом столбце. Мы предпочли их естественное расположение.

Если в нулевой строке есть отрицательные элементы, следует продолжить операции замещения, построить решение $x^{(3)}$ и т.д. На произвольном k-м шаге алгоритма симплекс-таблицу будем записывать в виде

$$\begin{array}{l|ccc|l} \text{Z} & -c'_1 & \cdots & -c'_{n+m} & c'_0 \\ \hline x_{B(1)} & a'_{11} & \cdots & a'_{1\,n+m} & b'_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{B(m)} & a'_{m1} & \cdots & a'_{m\,n+m} & b'_m \end{array} .$$

Здесь $J^{(k)} = \{B(1), ..., B(m)\}$ – МБН, соответствующее текущему базисному решению $x^{(k)}$, $B(i)$-й столбец таблицы равен $e_i \in E^{m+1}$, $i = 1, ..., m$, и целевая функция

$$Z = \sum_{j=1}^{n+m} c'_j x_j + c'_0$$

не зависит от базисных компонент. Если l-я строка – ведущая, то из базиса будет выведен $B(l)$-й столбец.

Алгоритм останавливается на k-м шаге, если решение $x^{(k)}$ удовлетворяет условию оптимальности, т.е. все элементы нулевой строки (последний элемент c'_0 не в счет) неотрицательны. В этом случае $x^{(k)}$ – оптимальное решение задачи (1.2), а $c'_0 = Z(x^{(k)})$ – оптимальное значение целевой функции. Возможен случай, когда $-c'_p < 0$, но $a'_{ip} \leqslant 0$, $i = 1, ..., m$. Как уже отмечалось, в этом случае задача (1.2) не имеет оптимального решения. Далее алгоритм симплекс-метода будем более коротко называть симплекс-алгоритмом.

Пример 4.1. Решим задачу оптимизации структуры страхового портфеля (пример 1.1). Ведущий элемент симплекс-таблицы будем подчеркивать. Запишем начальную симплекс-таблицу

$$\begin{array}{c|rrrrr|l} \text{Z} & -2 & -3 & 0 & 0 & 0 & 0 \\ \hline x_3 & 1 & 0 & 1 & 0 & 0 & 5 \\ x_4 & 0 & \underline{1} & 0 & 1 & 0 & 8 \\ x_5 & 4 & 3 & 0 & 0 & 1 & 32 \end{array} .$$

Здесь $x^{(1)} = (0, 0, 5, 8, 32)$ – начальное допустимое базисное решение. Переменные x_1, x_2 – небазисные. В нулевой строке элементы первых двух столбцов отрицательны и любой из этих столбцов может быть ведущим. Выберем ведущим второй столбец, поскольку целевая функция по переменной x_2 возрастает быстрее, чем по переменной x_1. Величина $\theta = \min[8/1, 32/3] = 8$ и вторая строка является ведущей. После преобразований получаем таблицу

$$\begin{array}{c|rrrrr|l} \text{Z} & -2 & 0 & 0 & 3 & 0 & 24 \\ \hline x_3 & 1 & 0 & 1 & 0 & 0 & 5 \\ x_2 & 0 & 1 & 0 & 1 & 0 & 8 \\ x_5 & \underline{4} & 0 & 0 & -3 & 1 & 8 \end{array} .$$

Из таблицы видно, что $x^{(2)} = (0, 8, 5, 0, 8)$. В нулевой строке находится единственный отрицательный элемент и первый столбец становится ведущим. Среди строк ведущая – третья.

После преобразований получим

Z	0	0	0	3/2	1/2	28
x_3	0	0	1	3/4	−1/4	3
x_2	0	1	0	1	0	8
x_1	1	0	0	−3/4	1/4	2

.

Нулевая строка таблицы содержит неотрицательные элементы. Поэтому $x^{(3)} = (2, 8, 3, 0, 0)$ – оптимальное решение задачи (1.2), а $x^* = (2, 8)$ – оптимальное решение исходной задачи (1.1).

Таким образом, страховая компания должна ориентироваться на заключение 20 тыс. договоров по первому виду страхования и 80 тыс. договоров по второму. При этом премия составит 280 тыс. условных единиц. Следует отметить, что при реализации оптимального решения компании необходимо иметь некоторый запас как по числу мест в больнице, так и по количеству операций. Действительно, увеличение числа заболеваний свыше средних значений может привести к нарушению ограничений.

Заметим, что симплекс-таблицу можно использовать в экономной форме, сохраняя только небазисные столбцы и их номера. Рассмотрим такую сокращенную симплекс-таблицу на произвольном k-м шаге алгоритма

	$x_{N(1)}$	$\cdots$	$x_{N(n)}$	
Z	$-c'_1$	$\cdots$	$-c'_n$	c'_0
$x_{B(1)}$	a'_{11}	$\cdots$	a'_{1n}	b'_1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$x_{B(m)}$	a'_{m1}	$\cdots$	a'_{mn}	b'_m

,

где $N(1), ..., N(n)$ – номера текущих небазисных столбцов. Преобразуем эту таблицу по симплекс-алгоритму. Пусть a'_{lp} – ведущий элемент. Тогда $B(l)$-й столбец в базисе будет заменен $N(p)$-м столбцом. Добавим в таблицу $(n+1)$-й единичный столбец e_l с номером $B(l)$ и проведем операцию замещения. Вычеркнем p-й столбец, а в нулевом столбце переменную $x_{B(l)}$ заменим на $x_{N(p)}$. В результате получим преобразованную симплекс-таблицу.

При большом числе ограничений использование сокращенных симплекс-таблиц приводит к значительной экономии памяти компьютера.

Пример 4.2. Решим симплекс-методом следующую задачу ЛП:

$$2x_1 + 3x_2 \to max$$

$$x_1 + 3x_2 \leqslant 2, \;\; 4x_1 + 2x_2 \leqslant 5, \;\; 2x_1 + x_2 \leqslant 4, \;\; x_1, x_2 \geqslant 0.$$

Выпишем последовательность сокращенных симплекс-таблиц, возникающих при использовании симплекс-метода:

	x_1	x_2	x_4	
Z	−2	−3	0	0
x_3	1	3	0	2
x_4	$\underline{4}$	2	1	5
x_5	2	1	0	4

,

	x_2	x_4	x_3	
Z	−2	1/2	0	5/2
x_3	$\underline{5/2}$	−1/4	1	3/4
x_1	1/2	1/4	0	5/4
x_5	0	−1/2	0	3/2

,

	x_4	x_3	
Z	$3/10$	$4/5$	$31/10$
x_2	$-1/10$	$2/5$	$3/10$
x_1	$3/10$	$-1/5$	$11/10$
x_5	$-1/2$	0	$3/2$

Из заключительной симплекс-таблицы находим оптимальное решение $x^* = (11/10, 3/10)$ стоимости 31/10.

§5. Преодоление зацикливания

Как уже отмечалось, если на некотором k-м шаге симплекс-алгоритма величина $\theta = 0$, то $x^{(k+1)} = x^{(k)}$, но базис при этом меняется. Здесь возможно зацикливание алгоритма.

Преодолеть зацикливание можно с помощью правила Блэнда. Пусть текущая симплекс-таблица имеет вид

Z	$-c'_1$	$\cdots$	$-c'_{n+m}$	c'_0
$x_{B(1)}$	a'_{11}	$\cdots$	$a'_{1\,n+m}$	b'_1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$x_{B(m)}$	a'_{m1}	$\cdots$	$a'_{m\,n+m}$	b'_m

Здесь $\{B(1), ..., B(m)\}$ – текущее МБН. Сформулируем правило Блэнда. Номер p ведущего столбца выбирается по формуле $p = \min\{j \mid -c'_j < 0\}$, а номер l ведущей строки – из условия

$$B(l) = \min_{k \in I} B(k), \quad \text{где} \quad I = \operatorname{Arg} \min_{i: a'_{ip} > 0} \frac{b'_i}{a'_{ip}}.$$

Отметим общую закономерность правила Блэнда: вводимый в базис p-й и выводимый из базиса $B(l)$-й столбцы в случае неоднозначного выбора имеют наименьшие возможные номера.

Теорема 5.1. При использовании правила Блэнда симплекс-алгоритм сходится за конечное число шагов.

§6. Методы поиска базисного допустимого решения

Если в задаче (1.2) некоторые величины b_i – отрицательные, то базисное решение $x^{(1)} = (\underbrace{0, ..., 0}_{n \text{ раз}}, b_1, ..., b_m)$ не является допустимым, что является затруднением для начала работы симплекс-метода. Аналогичное затруднение возникает в случае, когда задача ЛП задана в какой-либо другой форме. Здесь мы рассмотрим два метода построения начального БДР. Первый основан на введении так называемых *искусственных* переменных. Второй метод не требует новых переменных и использует только операции замещения. В конце параграфа будет показано, как можно с помощью операций замещения исключить свободные переменные.

Метод искусственного базиса

Пусть в исходной задаче ЛП все коэффициенты b_i неотрицательны, но в основных ограничениях с $(k+1)$-го по m-е фигурируют равенства. Вводя слабые переменные $x_{n+i},\ i = 1,...,k$, рассмотрим задачу ЛП

$$\begin{gathered} Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max \\ \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i,\ i = 1,...,k, \\ \sum_{j=1}^{n} a_{ij} x_j = b_i, \ \ i = k+1,...,m. \\ x_j \geqslant 0,\ j = 1,...,n+k. \end{gathered} \tag{6.1}$$

Эта задача отличается от задачи (1.2) тем, что в равенствах, начиная с $(k+1)$-го, отсутствуют слабые переменные. Поэтому в задаче (6.1) поиск начального БДР может вызвать затруднения. Чтобы их обойти, введем *искусственные* переменные

$$s_{n+i},\ i = k+1,...,m.$$

Они служат для приведения задачи к форме (1.2).

Рассмотрим вспомогательную задачу

$$\begin{gathered} U = -\sum_{i=k+1}^{m} s_{n+i} \to max \\ \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i,\ i = 1,...,k, \\ \sum_{j=1}^{n} a_{ij} x_j + s_{n+i} = b_i \ \ i = k+1,...,m, \\ x_j \geqslant 0,\ j = 1,...,n+k, \quad s_{n+i} \geqslant 0,\ i = k+1,...,m. \end{gathered} \tag{6.2}$$

Пусть X_2 – множество допустимых решений

$$\overline{x} = (x_1, ..., x_{n+k}, s_{n+k+1}, ..., s_{n+m})$$

задачи (6.2). Тогда решения из множества

$$X_1 = \{x \in X_2 \mid s_{n+i} = 0,\ i = k+1,...,m\}$$

соответствуют множеству допустимых решений задачи (6.1).

Поскольку в задаче (6.2) целевая функция ограничена сверху и множество допустимых решений X_2 не пусто, симплекс-алгоритм построит оптимальное БДР x^0 задачи (6.2). Если хотя бы одна из искусственных переменных $s^0_{n+i} > 0$, то множество X_1 пусто, а если $s^0_{n+i} = 0,\ i = k+1,...,m$, то $(x^0_1,...,x^0_{n+k})$ – БДР задачи (6.1). Можно продолжить симплекс-алгоритм для задачи (6.1), взяв $(x^0_1,...,x^0_{n+k})$ в качестве начального БДР. При этом столбцы, отвечающие искусственным переменным, из симплекс-таблицы вычеркиваются. Могут возникнуть следующие два случая:

1) Все искусственные переменные $s_{n+i},\ i = k+1, ..., m$, не являются базисными для решения x^0 задачи (6.2). Тогда симплекс-таблица не требует дополнительных преобразований и можно применить симплекс-метод к исходной задаче (6.1).

2) Хотя бы одна из переменных $s_{n+i},\ i = k+1, ..., m$, является базисной для решения x^0. Пусть без потери общности искусственная переменная s_{n+m} является базисной и находится в последней строке симплекс-таблицы, которой соответствует уравнение

$$\sum_{j=1}^{n+k} a'_{mj} x_j + \sum_{j=n+k+1}^{n+m-1} a'_{mj} s_j + s_{m+n} = 0.$$

Будем считать, что найдется хотя бы один коэффициент $a'_{mp} \neq 0$, где $p \leqslant m + n - 1$. Действительно, если $a'_{mj} = 0,\ j = 1, ..., n+m-1$, то m-ю строку симплекс-таблицы можно вычеркнуть без изменения множества допустимых решений. Ясно, что p-й столбец не принадлежит базису. После операции замещения с ведущим элементом a'_{mp} $(m+n)$-й столбец симплекс-таблицы, соответствующий переменной s_{n+m}, будет заменен в базисе p-м столбцом. Эту операцию можно повторить и для других базисных искусственных переменных.

Пример 6.1. Рассмотрим задачу

$$Z = 3x_1 + x_2 \quad \to \quad max$$

$$x_1 + x_2 \leqslant 2,\ -x_1 + x_2 = 2,\ x_1, x_2 \geqslant 0.$$

После введения слабой переменной x_3 получаем задачу:

$$\begin{gathered} Z = 2x_1 + 3x_2 \quad \to \quad max \\ x_1 + x_2 + x_3 = 2,\ -x_1 + x_2 = 2,\ \ x_1, x_2, x_3 \geqslant 0. \end{gathered} \tag{6.1}$$

Рассмотрим вспомогательную задачу (6.2), введя искусственную переменную s_4 и изменив целевую функцию:

$$\begin{gathered} U = -s_4 \quad \to \quad max \\ x_1 + x_2 + x_3 = 2,\ -x_1 + x_2 + s_4 = 2, \\ x_1, x_2, x_3, s_4 \geqslant 0. \end{gathered} \tag{6.2}$$

Задачу (6.2) будем решать симплекс-методом; при этом будем использовать две строки для функций U и Z. Выпишем начальную симплекс-таблицу

$$\begin{array}{c|cccc|c} U & 0 & 0 & 0 & 1 & 0 \\ Z & -3 & -1 & 0 & 0 & 0 \\ \hline x_3 & 1 & \underline{1} & 1 & 0 & 2 \\ s_4 & -1 & 1 & 0 & 1 & 2 \end{array} \ .$$

Строка таблицы, соответствующая U, содержит число 1. Для начального базисного решения $x^{(1)} = (0, 0, 2, 2)$ не выполнено условие б) свойства (α) из § 4. Поэтому сначала преобразуем строку, соответствующую целевой функции U:

$$\begin{array}{c|cccc|c} U & 1 & -1 & 0 & 0 & -2 \\ Z & -3 & -1 & 0 & 0 & 0 \\ \hline x_3 & 1 & \underline{1} & 1 & 0 & 2 \\ s_4 & -1 & 1 & 0 & 1 & 2 \end{array} \ .$$

Следующая таблица получена после итерации по симплекс-алгоритму для задачи (6.2):

$$\begin{array}{c|rrrr|c} U & 2 & 0 & 1 & 0 & 0 \\ Z & -2 & 0 & 1 & 0 & 2 \\ \hline x_2 & 1 & 1 & 1 & 0 & 2 \\ s_4 & -2 & 0 & \underline{-1} & 1 & 0 \end{array}\ .$$

Отсюда находим, что $x^0 = (0,2,0,0)$ – оптимальное решение задачи (6.2). Поскольку $s_4^0 = 0$, то базисное решение – вырожденное. Вычеркнем строку, соответствующую U, и столбец, соответствующий искусственной переменной s_4. Осуществим операцию замещения с ведущим элементом a_{23}:

$$\begin{array}{c|rrr|c} Z & -4 & 0 & 0 & 2 \\ \hline x_2 & -1 & 1 & 0 & 2 \\ x_3 & \underline{2} & 0 & 1 & 0 \end{array}\ .$$

Затем продолжим симплекс-метод для задачи (6.1) с целевой функцией Z. В результате получим таблицу

$$\begin{array}{c|ccc|c} Z & 0 & 0 & 2 & 2 \\ \hline x_2 & 0 & 1 & 1/2 & 2 \\ x_1 & 1 & 0 & 1/2 & 0 \end{array}\ .$$

Отсюда $x^* = (0,2)$ – оптимальное решение исходной задачи.

Пусть в задаче (1.2) имеется коэффициент $b_i < 0$. Тогда следует умножить i-е уравнение на -1 и применить метод «искусственного базиса», добавив искусственную переменную s_{n+i}. В результате получится ограничение

$$\sum_{j=1}^{n}(-a_{ij})x_j - x_{n+i} + s_{n+i} = -b_i.$$

Метод замещений

Пусть в задаче (1.2) имеется коэффициент $b_s < 0$. Тогда базисное решение $x^{(1)} = (\underbrace{0,...,0}_{n \text{ раз}}, b_1, ..., b_m)$ не является допустимым. В процессе приводимого ниже алгоритма, начиная с $x^{(1)}$, осуществляется переход от одного базисного решения к другому. При этом неотрицательные компоненты решений сохраняют знак, а по крайней мере одна из отрицательных компонент возрастает.

Если элементы s-й строки симплекс-таблицы $a_{sj},\ j = 1, ..., n$, неотрицательны, то допустимого решения задачи (1.2) не существует, поскольку $b_s < 0$. Пусть найдется такой номер p, что $a_{sp} < 0$. Тогда p-й столбец – небазисный. Выберем l-ю строку из условия

$$\overline{\theta} = \min_{i \in I(p)} b_i/a_{ip} = b_l/a_{lp},$$

где $I(p) = \{i \mid a_{ip} < 0,\ b_i < 0\} \cup \{i \mid a_{ip} > 0,\ b_i \geqslant 0\}$. Возьмем ведущим элемент a_{lp} и сделаем операцию замещения. После преобразований компоненты последнего столбца симплекс-таблицы станут равными

$$b_i^1 = \begin{cases} b_l/a_{lp}\,, & i = l, \\ b_i - b_l a_{ip}/a_{lp}\,, & i \neq l. \end{cases}$$

Покажем, что если $b_i \geqslant 0$, то и $b_i^1 \geqslant 0$. Действительно, по определению $\overline{\theta} = b_l/a_{lp} \geqslant 0$ и при $a_{ip} \leqslant 0$ утверждение очевидно. Если $a_{ip} > 0$, то из неравенства $b_i/a_{ip} \geqslant b_l/a_{lp}$ следует, что $b_i^1 \geqslant 0$. Итак, неотрицательные элементы последнего столбца симплекс-таблицы после операции замещения сохраняют свой знак.

Если $s = l$, то $b_s^1 = b_s/a_{sp} > 0$ и число отрицательных элементов последнего столбца уменьшится. Пусть $s \neq l$. Тогда $b_s^1 = b_s - b_l a_{sp}/a_{lp} \geqslant b_s$. Итак, s-я компонента последнего столбца симплекс-таблицы не убывает. Операции замещения продолжаются до тех пор, пока на некотором k-м шаге элемент b_s^k не станет неотрицательным. Если при некотором $s_1 \neq s$ $b_{s_1}^k < 0$, то нужно продолжить операции замещения и добиться, чтобы на некотором шаге k_1 $b_{s_1}^{k_1} \geqslant 0$ и т.д. Алгоритм заканчивает свою работу, когда все элементы последнего столбца симплекс-таблицы становятся неотрицательными. В результате будет построено БДР.

Отметим, что при $\overline{\theta} = 0$ возможно зацикливание алгоритма (см. упражнение 6.2). Для его преодоления можно использовать правило Блэнда. Действительно, при $\overline{\theta} = 0$ выполнено условие $a_{lp} > 0$, $b_l = 0$, $l \neq s$. Поэтому любая операция замещения в этом случае такая же, как и при работе симплекс-алгоритма, решающего задачу ЛП

$$x_{n+s} = -\sum_{j=1}^{n} a_{sj}x_j + b_s \to max$$

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i,\ i \neq s,\ \ x_j \geqslant 0,\ j = 1, ..., n+m.$$

Сформулируем правило Блэнда для метода замещений. Рассмотрим произвольный шаг алгоритма. Пусть $\{B(1), ..., B(m)\}$ – текущее МБН и $b'_s < 0$. Тогда $p = \min\{j \mid a'_{sj} < 0\}$ – номер ведущего столбца. Номер l ведущей строки определим из условия $B(l) = \min\limits_{k \in I} B(k)$, где $I = \operatorname{Arg} \max\limits_{i \in I(p)} b'_i/a'_{ip}$.

Пример 6.2. Пусть ограничения задачи ЛП заданы системой

$$-2x_1 + x_2 - x_3 = -2, \quad -x_1 - x_2 + 2x_3 = 3, \quad x_1, x_2, x_3 \geqslant 0.$$

Применим сначала к системе уравнений преобразования Гаусса-Жордана и приведем ее к диагональному виду относительно переменных x_2 и x_3:

$$\begin{array}{c|rrr|r} & & & & \\ \hline & -2 & 1 & -1 & -2 \\ & -1 & -1 & 2 & 3 \end{array} \sim \begin{array}{c|rrr|r} & & & & \\ \hline & -2 & 1 & -1 & -2 \\ & -3 & 0 & 1 & 1 \end{array} \sim \begin{array}{c|rrr|r} & & & & \\ \hline x_2 & -5 & 1 & 0 & -1 \\ x_3 & -3 & 0 & 1 & 1 \end{array}\ .$$

В полученной симплекс-таблице $a_{11} = -5$ – ведущий элемент. После операции замещения таблица приобретает требуемый вид

$$\begin{array}{c|ccc|c} & & & & \\ \hline x_1 & 1 & 1/5 & 0 & 1/5 \\ x_3 & 0 & -3/5 & 1 & 8/5 \end{array}\ .$$

Здесь $x^{(1)} = (1/5, 0, 8/5)$ – начальное БДР.

Исключение свободных переменных

Пусть задача ЛП приведена к виду

$$Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i,\ i = 1, ..., m,$$

где некоторые переменные являются свободными. Покажем, каким образом свободные переменные можно исключить, используя операции замещения. Рассмотрим начальную таблицу с МБН $J = \{n+1, ..., n+m\}$. Если базисная переменная x_{n+i} – свободная, то i-е уравнение и $(n+i)$-й столбец симплекс-таблицы можно вычеркнуть, поскольку целевая функция и все уравнения с номерами, отличными от i, не зависят от x_{n+i}. Пусть $(x_j^0,\ j \neq n+i)$ – оптимальное решение получившейся задачи. Тогда свободная компонента находится по формуле

$$x_{n+i}^0 = b_i - \sum_{j=1}^{n} a_{ij} x_j^0.$$

В результате получили $(x_j^0,\ j = 1, ..., n+m)$ – оптимальное решение исходной задачи. Так можно исключить все базисные свободные компоненты.

Пусть $x_p,\ p \leqslant n,$ – свободная небазисная переменная. Если $a_{ip} = 0,\ i = 1, ..., m,$ и $c_p = 0$, то p-й столбец можно вычеркнуть, поскольку переменная x_p не влияет ни на ограничения, ни на целевую функцию[1]. Если $a_{ip} = 0,\ i = 1, ..., m,$ и $c_p \neq 0$, то переменная x_p может принимать любые значения, целевая функция не ограничена сверху и задача не имеет оптимального решения. Предположим, что найдется элемент $a_{lp} \neq 0$. Тогда делаем операцию замещения, взяв этот элемент ведущим. В результате переменная x_p будет введена в базис и ее можно исключить указанным выше способом. За несколько итераций все свободные переменные будут исключены.

Пример 6.3. Рассмотрим задачу ЛП

$$Z = 2x_1 - x_2 + 3x_3 - 4x_4 \to max$$

$$x_1 - x_2 + x_3 + x_4 \leqslant -2,\ x_1 + 2x_2 - x_3 - 2x_4 \leqslant 1,$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_4 \geqslant 0,$$

[1]Это означает, что переменная x_p была введена в модель ошибочно.

в которой x_3 – свободная переменная. Введем слабые переменные x_5, x_6 и выпишем начальную симплекс-таблицу

Z	-2	1	-3	4	0	0	0
x_5	1	-1	$\underline{1}$	1	1	0	-2
x_6	1	2	-1	-2	0	1	1

.

Введем свободную переменную x_3 в базис. После операции замещения с ведущим элементом a_{13}, получим симплекс-таблицу

Z	1	-2	0	7	3	0	-6
x_3	1	-1	$\underline{1}$	1	1	0	-2
x_6	2	1	0	-1	1	1	-1

.

Вычеркнем первую строку

Z	1	-2	0	7	3	0	-6
x_6	2	1	0	$\underline{-1}$	1	1	-1

и выразим свободную переменную x_3 через остальные:

$$x_3 = -2 - x_1 + x_2 - x_4 - x_5.$$

Найдем начальное БДР. Для этого сделаем операцию замещения в полученной симплекс-таблице с ведущим элементом $a'_{14} = -1$:

Z	15	5	0	0	10	7	-13
x_4	-2	-1	0	1	-1	-1	1

.

Отсюда с учетом формулы для x_3 находим оптимальное решение $x^* = (0, 0, -3, 1)$.

§7. Двойственная задача

Напомним задачу ЛП в стандартной форме:

$$Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max$$
$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, ..., m, \qquad x_j \geqslant 0, \ j = 1, ..., n. \tag{1.1}$$

Эту задачу можно решать симплекс-методом. Если все b_i неотрицательны, то начальная симплекс-таблица называется *прямо допустимой* и в качестве начального БДР можно использовать $x^{(1)} = (\underbrace{0, ..., 0}_{n \text{ раз}}, b_1, ..., b_m)$. Если среди b_i есть отрицательные, то начальное БДР можно найти методом замещений. После его нахождения можно продолжать алгоритм с таблицей, строки которой являются линейными комбинациями строк начальной симплекс-таблицы

Z	$-c_1$	$\dots$	$-c_n$	0	$\dots$	0	0
x_{n+1}	a_{11}	$\dots$	a_{1n}	1	$\dots$	0	b_1
$\vdots$		$\vdots$			$\ddots$		$\vdots$
x_{n+m}	a_{m1}	$\dots$	a_{mn}	0	$\dots$	1	b_m

.

Определим задачу ЛП, двойственную к (1.1). Введем *двойственные* переменные. В процессе нахождения начального базисного решения и выполнения симплекс-алгоритма делаются элементарные преобразования над строками этой таблицы. При этом к нулевой строке добавляются другие строки, а нулевая строка не добавляется ни к каким другим строкам. Таким образом, на любом шаге алгоритма нулевая строка таблицы есть начальная нулевая строка плюс линейная комбинация остальных строк начальной таблицы и ее можно записать в виде

Z	$z_1 - c_1$	$\ldots$	$z_n - c_n$	y_1	$\ldots$	y_m	y_0

.

Здесь вектор $(z_1, ..., z_n, y_1, ..., y_m)$ представляет собой линейную комбинацию строк начальной таблицы с номерами $i \geqslant 1$. Ясно, что коэффициентами этой комбинации будут $y_1, ..., y_m$. Поэтому имеем

$$z_j = \sum_{i=1}^{m} a_{ij} y_i, \ j = 1, ..., n, \ \ y_0 = \sum_{i=1}^{m} b_i y_i,$$

где y_0 – значение целевой функции Z на текущем базисном решении. Переменные $y_1, ..., y_m$ будем называть двойственными.

Условие оптимальности в новых обозначениях имеет вид

$$z_j - c_j \geqslant 0, \ j = 1, ..., n, \ \ y_i \geqslant 0, \ i = 1, ..., m.$$

Из предыдущих соотношений следует, что оно эквивалентно неравенствам

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \ j = 1, ..., n, \ \ y_i \geqslant 0, \ i = 1, ..., m.$$

Получили ограничения на двойственные переменные. Эти ограничения выполнены только для заключительной таблицы симплекс-метода, в которой содержится оптимальное БДР. Пусть $y_1^*, ..., y_m^*$ – значения двойственных переменных в заключительной таблице. При этом $y_0^* = \sum\limits_{i=1}^{m} b_i y_i^*$ – наибольшее значение целевой функции задачи (1.1).

Из дальнейшего будет видно, что y_0^* – наименьшее значение функции $y_0 = \sum\limits_{i=1}^{m} b_i y_i$ при выполнении ограничений на двойственные переменные. Итак, возникает задача

$$\begin{gathered} y_0 = \sum_{i=1}^{m} b_i y_i \ \ \rightarrow \ \ min \\ \sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \ j = 1, ..., n, \ \ y_i \geqslant 0, \ i = 1, ..., m. \end{gathered} \tag{7.1}$$

Эта задача называется двойственной к «прямой» задаче (1.1) и имеет стандартную форму.

Рассмотрим соотношения двойственности. Для решений задач (1.1) и (7.1) будем использовать обозначения $x = (x_1, ..., x_n)$ и $y = (y_1, ..., y_m)$. Пусть X и Y – множества допустимых решений задач (1.1) и (7.1).

Теорема 7.1. Если множества X и Y не пусты, то для любых решений $x \in X$, $y \in Y$ выполнено неравенство

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i y_i,$$

где равенство достигается на оптимальных решениях задач (1.1) и (7.1).

Следствие 1.

а) Если $X \neq \emptyset$ и функция Z не ограничена сверху, то $Y = \emptyset$.

б) Если $Y \neq \emptyset$ и функция y_0 не ограничена снизу, то $X = \emptyset$.

в) $X \neq \emptyset$ и функция Z ограничена сверху тогда и только тогда, когда $Y \neq \emptyset$ и функция y_0 ограничена снизу.

Замечание. В последнем случае обе задачи (прямая и двойственная) имеют оптимальные решения. Достаточно, чтобы одна задача имела оптимальное решение, тогда и другая задача также будет иметь оптимальное решение.

Следствие 2 (Свойство дополняющей нежесткости). Пусть x^* и y^* – допустимые решения задач (1.1) и (7.1). Для того чтобы x^* и y^* были оптимальными решениями, необходимо и достаточно, чтобы выполнялись равенства

1) $x_j^*(\sum\limits_{i=1}^{m} a_{ij} y_i^* - c_j) = 0,\ j = 1, ..., n;$

2) $y_i^*(b_i - \sum\limits_{j=1}^{n} a_{ij} x_j^*) = 0,\ i = 1, ..., m.$

Поясним термин «дополняющая нежесткость». Если в 1) или 2) один из множителей положителен, то другой обязательно равен нулю. Например, если $x_j^* > 0$ (ограничение $x_j \geqslant 0$ выполнено «нежестко»), то $\sum\limits_{i=1}^{m} a_{ij} y_i^* - c_j = 0$ (соответствующее ограничение $\sum\limits_{i=1}^{m} a_{ij} y_i \geqslant c_j$ двойственной задачи выполнено «жестко»).

Пример 7.1. Используя теорию двойственности решим следующую задачу ЛП:

$$\begin{gathered} Z = 12x_1 + 18x_2 + 8x_3 \to max \\ 4x_1 + 3x_2 + x_3 \leqslant 2,\ \ (y_1) \\ x_1 + 2x_2 + x_3 \leqslant 1,\ \ (y_2) \\ x_1, x_2, x_3 \geqslant 0. \end{gathered}$$

Справа от основных ограничений указаны соответствующие им двойственные переменные. Выпишем двойственную задачу

$$\begin{gathered} y_0 = 2y_1 + y_2 \to min \\ 4y_1 + y_2 \geqslant 12, \quad (x_1) \\ 3y_1 + 2y_2 \geqslant 18, \quad (x_2) \\ y_1 + y_2 \geqslant 8, \qquad (x_3) \\ y_1, y_2 \geqslant 0. \end{gathered}$$

Множество ее допустимых решений имеет вершины $y^{(1)}=(0,12),\ y^{(2)}=(6/5,36/5)$, $y^{(3)}=(2,6),\ y^{(4)}=(8,0)$ (рис. 7.1). При этом $y^{(2)}$ – оптимальное решение двойственной задачи.

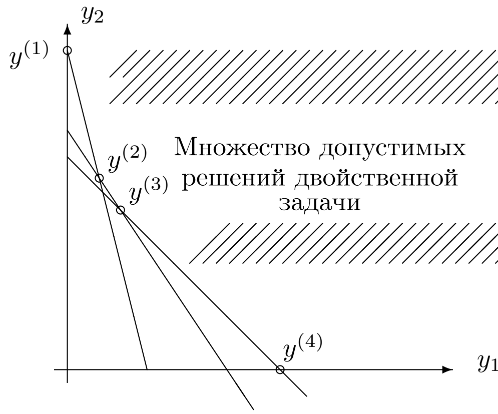

Рис. 7.1

Поскольку $y_1^{(2)}, y_2^{(2)} > 0,\ y_1^{(2)} + y_2^{(2)} > 8$, то по свойству дополняющей нежесткости оптимальное решение x^0 прямой задачи удовлетворяет системе

$$4x_1 + 3x_2 + x_3 = 2, \ \ x_1 + 2x_2 + x_3 = 1, \ \ x_3 = 0.$$

Отсюда $x^0 = (1/5, 2/5, 0)$.

Дадим интерпретацию двойственным переменным. Рассмотрим (1.1) как задачу планирования производства. Пусть x^* и y^* – оптимальные решения задач (1.1)и (7.1). Предположим, что для оптимального плана производства x^* ресурс i-го типа используется не полностью, т.е. выполнено неравенство $\sum\limits_{j=1}^{n} a_{ij}x_j^* < b_i$. Тогда по свойству дополняющей нежесткости $y_i^* = 0$. Это наводит на мысль рассматривать число y_i^* как оценку дефицитности ресурса i-го типа. Действительно, раз в оптимальном плане x^* ресурс используется не полностью, то он не дефицитен и его оценка равна нулю. Более того, если увеличить значение b_i, то решения x^* и y^* будут по-прежнему оптимальными в задачах (1.1) и (7.1), поскольку они остаются допустимыми и для них выполнено свойство дополняющей нежесткости (см. следствие 2). Поэтому при увеличении количества ресурса b_i максимум целевой функции в задаче (1.1)

$$M(b_1, ..., b_m) \stackrel{def}{=} \max_{x\in X} \sum_{j=1}^{n} c_j x_j$$

остается постоянным.

Пусть теперь $y_i^* > 0$. Тогда по свойству дополняющей нежесткости $\sum\limits_{j=1}^{n} a_{ij}x_j^* = b_i$ и ресурс i-го типа дефицитен, так как при реализации плана производства x^* используется полностью. Посмотрим, как зависит функция $M(b_1, ..., b_m)$ от переменной b_i. Предположим, что при любых $b_i > 0,\ i = 1, ..., m$, задача (1.1) имеет оптимальное решение. Для этого достаточно потребовать, чтобы матрица A содержала неотрицательные элементы и в каждой ее строке имелся положительный элемент. По теореме 7.1 функцию M можно записать в виде

$$M(b_1, ..., b_m) = \min_{y\in Y} \sum_{i=1}^{m} b_i y_i = \min_{y\in Y^0} \sum_{i=1}^{m} b_i y_i,$$

где Y^0 – множество вершин множества Y. Поскольку Y^0 – конечное множество, то функция $M(b_1,...,b_m)$ непрерывна. Пусть для вектора ресурсов $(b_1,...,b_m)$ задача (7.1) имеет единственное оптимальное решение y^*. Тогда из непрерывности функции M вытекает, что при малом Δb_i для вектора ресурсов $(b_1,...,b_i+\Delta b_i,...,b_m)$ задача (7.1) имеет то же самое оптимальное решение y^*. Следовательно,

$$\Delta M_i \stackrel{def}{=} M(b_1,...,b_i+\Delta b_i,...,b_m) - M(b_1,...,b_m) = \Delta b_i y_i^*.$$

Таким образом, в сделанных предположениях y_i^* является скоростью возрастания максимума целевой функции задачи (1.1) при увеличении ресурса i-го типа. Ясно, что чем больше значение y_i^*, тем более дефицитен ресурс.

§8. Двойственный симплекс-метод

Симплекс-метод, примененный к двойственной задаче, называется двойственным. Его особенность заключается в том, что преобразования осуществляются с симплекс-таблицей прямой задачи. Рассмотрим прямую задачу ЛП в диагональной форме

$$Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i, \ i = 1,...,m, \tag{1.2}$$

$$x_j \geqslant 0, \ j = 1,...,n+m.$$

Двойственную задачу (7.1) также можно записать в форме равенств:

$$y_0 = \sum_{i=1}^{m} b_i y_i \quad \to \quad min$$

$$\sum_{i=1}^{m} a_{ij} y_i - y_{m+j} = c_j, \ j = 1,...,n, \tag{8.1}$$

$$y_i \geqslant 0, \ i = 1,...,m+n.$$

Между переменными задач (1.2) и (8.1) имеется следующее взаимно однозначное соответствие:

x_1	$\dots$	x_n	x_{n+1}	$\dots$	x_{n+m}
y_{m+1}	$\dots$	y_{m+n}	y_1	$\dots$	y_n

.

Действительно, рассмотрим нулевую строку симплекс-таблицы на произвольном шаге алгоритма симплекс-метода

Z	$z_1 - c_1$	$\dots$	$z_n - c_n$	y_1	$\dots$	y_m	y_0

,

где $z_j = \sum\limits_{i=1}^{m} a_{ij} y_i$. Из ограничений задачи (8.1) вытекает, что

$$z_j - c_j = y_{m+j},\ j = 1, ..., n.$$

Поэтому нулевая строка симплекс-таблицы имеет вид

$$\begin{array}{c|cccccc|c} Z & y_{m+1} & \dots & y_{m+n} & y_1 & \dots & y_m & y_0 \\ \hline & & & & & & & \end{array}.$$

Столбцы этой таблицы отвечают переменным $x_1, ..., x_{n+m}$. Отсюда и получаем искомое соответствие. Отметим также, что основным переменным задачи (1.2) соответствуют слабые переменные задачи (8.1) и наоборот. Учитывая это, можно переформулировать свойство дополняющей нежесткости для оптимальных решений задач (1.2) и (8.1) (см. следствие 2 теоремы 7.1):

Свойство дополняющей нежесткости. Допустимые решения $(x_1^*, ..., x_{n+m}^*)$ и $(y_1^*, ..., y_{n+m}^*)$ задач (1.2) и (8.1) тогда и только тогда оптимальны, когда справедливы равенства

1) $x_j^* y_{m+j}^* = 0,\ j = 1, ..., n$;
2) $y_i^* x_{n+i}^* = 0,\ i = 1, ..., m$.

Теперь рассмотрим начальную симплекс-таблицу

$$\begin{array}{c|cccccc|c} Z & -c_1 & \dots & -c_n & 0 & \dots & 0 & 0 \\ \hline x_{n+1} & a_{11} & \dots & a_{1n} & 1 & \dots & 0 & b_1 \\ \vdots & \vdots & & & \vdots & & & \vdots \\ x_{n+m} & a_{m1} & \dots & a_{mn} & 0 & \dots & 1 & b_m \end{array}.$$

Для применения двойственного симплекс-метода необходимо, чтобы ее нулевая строка содержала неотрицательные элементы, т.е. $-c_j \geqslant 0,\ j = 1, ..., n$. Такая симплекс-таблица называется *двойственно допустимой.* В этом случае

$$y^{(1)} = (\underbrace{0, ..., 0}_{m \text{ раз}}, -c_1, ..., -c_n)$$

– допустимое решение задачи (8.1). Нетрудно видеть, что это – БДР, поскольку столбцы матрицы ограничений, соответствующие последним n компонентам образуют матрицу $-I_n$ и линейно независимы. Итак, для решения $y^{(1)}$ множество $I^{(1)} = \{m+1, ..., m+n\}$ является МБН. Базисным переменным $y_{m+1}, ..., y_{m+n}$ задачи (8.1) отвечают переменные $x_1, ..., x_n$ задачи (1.2). Оставшиеся переменные $x_{n+1}, ..., x_{n+m}$ являются базисными переменными решения

$$x^{(1)} = (\underbrace{0, ..., 0}_{n \text{ раз}}, b_1, ..., b_m)$$

прямой задачи. Таким образом, МБН решений $x^{(1)}$ и $y^{(1)}$ взаимно дополняют друг друга. Это свойство будет сохранено и на последующих шагах алгоритма.

Отметим свойство $(\alpha)'$ базисного решения $y^{(1)}$ по отношению к симплекс-таблице прямой задачи:

а)$'$ Столбцы таблицы, отвечающие небазисным компонентам решения $y^{(1)}$, образуют единичную матрицу.

б)$'$ Целевая функция $y_0 = \sum\limits_{i=1}^{m} b_i y_i$ зависит только от небазисных переменных.

Свойство $(\alpha)'$ будет поддерживаться в процессе двойственного симплекс-алгоритма. Из пункта б)$'$ вытекает, что $y_0(y^{(1)}) = 0$.

Теперь перейдем от решения $y^{(1)}$ к другому БДР $y^{(2)}$ с меньшим значением целевой функции. С этой целью возьмем небазисную компоненту $y_l,\ l \leqslant m$, и будем ее увеличивать от нуля. В результате получим решение вида

$$\tilde{y} = (0, ..., 0, y_l, 0, ..., 0, y_{m+1}, ..., y_{m+n}).$$

Для того чтобы это решение было допустимым в задаче (8.1), необходимо, чтобы

$$y_{m+j} = a_{lj} y_l - c_j \geqslant 0,\ j = 1, ..., n.$$

Отметим, что $y_0(\tilde{y}) = b_l y_l$. Поэтому целевая функция при увеличении y_l убывает, если $b_l < 0$. Чтобы в нулевой строке симплекс-таблицы получить вектор $\tilde{y}$, необходимо l-ю строку умножить на y_l и добавить ее к нулевой строке. l-я строка, как и ранее, называется ведущей. Итак, выбор ведущей строки определяется условием $b_l < 0$.

Если $a_{lj} \geqslant 0,\ j = 1, ..., n$, то решение $\tilde{y}$ будет допустимым при любом значении $y_l > 0$. Следовательно, при увеличении y_l целевая функция y_0 на решении $\tilde{y}$ будет неограниченно убывать и задача (8.1) не имеет оптимального решения. В этом случае двойственный симплекс-алгоритм заканчивает работу.

Предположим, что в l-й строке найдется элемент $a_{lj} < 0$. Тогда решение $\tilde{y}$ будет допустимым, если $y_l \leqslant c_j / a_{lj}$ для всех j, для которых $a_{lj} < 0$. Отсюда получаем, что

$$y_l \leqslant \theta' = \min_{j: a_{lj} < 0} \frac{-c_j}{-a_{lj}} = \frac{-c_p}{-a_{lp}}.$$

При $y_l = c_p / a_{lp}$ получим, что $y_{m+p} = 0$ и переменная y_{m+p} будет выведена из базиса, а переменная y_l – введена в базис. Чтобы получить $y_{m+p} = 0$, сделаем операцию замещения с ведущим элементом a_{lp}. В результате в нулевой строке таблицы получим БДР

$$y^{(2)} = (0, ..., 0, \underset{l}{\theta'}, 0, ..., 0, a_{l1}\theta' - c_1, ..., \underset{m+p}{0}, ..., a_{ln}\theta' - c_n).$$

Отметим, что столбцы симплекс-таблицы, отвечающие небазисным переменным $y_i,\ i \neq l,\ y_{m+p}$, будут в преобразованной таблице базисными для $x^{(2)}$. Далее, если для преобразованной таблицы выписать двойственную задачу (эквивалентную исходной), то в ней целевая функция будет зависеть только от переменных $y_i,\ i \neq l,\ y_{m+p}$. Тем самым, свойство $(\alpha)'$ сохраняется.

Алгоритм останавливает работу на k-м шаге, когда в текущей симплекс-таблице выполнено одно из следующих двух условий:

1) $b'_i \geqslant 0,\ i = 1, ..., m$. В этом случае текущие базисные решения $x^{(k)}$ и $y^{(k)}$ оптимальны в задачах (1.2) и (8.1).

2) В ведущей l-й строке все элементы a'_{lj} неотрицательны. В этом случае целевая функция задачи (8.1) не ограничена снизу.

Пример 8.1. Рассмотрим задачу ЛП

$$2x_1 + x_2 \;\; \to \;\; min$$

$$x_1 - 2x_2 \leqslant 2, \;\; 3x_1 - x_2 \geqslant 1, \;\; x_1, x_2 \geqslant 0.$$

Преобразуем ее к форме (1.2):

$$Z = -2x_1 - x_2 \;\; \to \;\; max$$

$$x_1 - 2x_2 + x_3 = 2, \;\; -3x_1 + x_2 + x_4 = -1, \;\; x_1, x_2, x_3, x_4 \geqslant 0.$$

Начальная симплекс-таблица имеет вид

$$\begin{array}{c|rrrr|r} Z & 2 & 1 & 0 & 0 & 0 \\ \hline x_3 & 1 & -2 & 1 & 0 & 2 \\ x_4 & \underline{-3} & 1 & 0 & 1 & -1 \end{array} \; .$$

Здесь нулевая строка неотрицательна. Поэтому $y^{(1)} = (0, 0, 2, 1)$ – начальное базисное решение. Ведущие вторая строка и первый столбец определяются однозначно. После преобразования таблицы получим

$$\begin{array}{c|rrrr|r} & y_3 & y_4 & y_1 & y_2 & \\ Z & 0 & 5/3 & 0 & 2/3 & -2/3 \\ \hline x_3 & 0 & -5/3 & 1 & 1/3 & 5/3 \\ x_1 & 1 & -1/3 & 0 & -1/3 & 1/3 \end{array} \; .$$

Поскольку $b'_i > 0, \; i = 1, 2$, то $y^{(2)} = (0, 2/3, 0, 5/3)$ – оптимальное решение задачи (8.1), $x^{(2)} = (1/3, 0, 5/3, 0)$ – оптимальное решение задачи (1.2), а $x^* = (1/3, 0)$ – оптимальное решение исходной задачи ЛП. Отметим, что x^* – единственное оптимальное решение. Действительно, в оптимальном решении двойственной задачи $y_4^* > 0, \; y_2^* > 0$. Следовательно, по свойству дополняющей нежесткости в любом оптимальном решении прямой задачи (1.2) соответствующие вторая и четвертая компоненты будут нулевыми.

Возможное зацикливание двойственного симплекс-алгоритма можно устранить, используя правило выбора ведущего элемента, аналогичное правилу Блэнда для прямого симплекс-метода.

§9. Задача ЛП с двусторонними ограничениями

Рассмотрим задачу

$$\begin{gathered} Z = \sum_{j=1}^{n} c_j x_j \;\; \to \;\; max \\ \sum_{j=1}^{n} a_{ij} x_j \;\; \leqslant b_i, \; i = 1, ..., m, \;\; 0 \leqslant x_j \leqslant u_j, \; j = 1, ..., n. \end{gathered} \tag{9.1}$$

По сравнению с задачей (1.1) здесь на переменные наложены дополнительные ограничения $x_j \leqslant u_j$, где $u_j, \; j = 1, ..., n$ – заданные положительные числа. Подобные задачи с двусторонними ограничениями возникают, например, в тех случаях, когда исходным множеством решений является параллелепипед. Встретятся они нам и при решении задач целочисленного ЛП методом ветвей и границ.

Сформулируем задачу с двусторонними ограничениями в диагональной форме:

$$Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i,\ i = 1, ..., m, \tag{9.2}$$

$$0 \leqslant x_j \leqslant u_j,\ j = 1, ..., n,\ x_{n+i} \geqslant 0,\ i = 1, ..., m.$$

Отметим, что на слабые переменные x_{n+i} не накладываются двусторонние ограничения. Можно было ввести дополнительные слабые переменные и превратить неравенства $x_j \leqslant u_j$ в равенства, но при этом возросла бы размерность симплекс-таблицы. Оказывается этого можно не делать и работать с обычной симплекс-таблицей задачи (1.2).

В дальнейшем будем предполагать, что $b_i \geqslant 0,\ i = 1, ..., m$. Как и при решении задачи (1.2), это дает возможность начать работу симплекс-алгоритма с БДР $x^{(1)} = (\underbrace{0, ..., 0}_{n\ \text{раз}}, b_1, ..., b_m)$.

Наряду с переменными $x_j,\ j = 1, ..., n$, будем рассматривать *дополняющие* переменные $y_j = u_j - x_j,\ j = 1, ..., n$. Очевидно, что $0 \leqslant y_j \leqslant u_j$. Если в задаче (1.2) какие-то переменные x_j заменяются на $u_j - y_j$, то получается эквивалентная задача. Этим преобразованием будем пользоваться в процессе применения симплекс-метода. Посмотрим подробнее, как преобразуется симплекс-таблица при замене переменной $x_j = u_j - y_j$.

Рассмотрим симплекс-таблицу на произвольном шаге алгоритма

Z	$-c'_1$	$\dots$	$-c'_{n+m}$	c'_o
$x_{B(1)}$	a'_{11}	$\dots$	a'_{1n+m}	b'_1
$\vdots$		$\vdots$		$\vdots$
$x_{B(m)}$	a'_{m1}	$\dots$	a'_{mn+m}	b'_m .

Замена переменной x_j на y_j приводит к тому, что из последнего столбца симплекс-таблицы вычитается j-й, умноженный на u_j, а элементы j-го столбца меняют знак.

Обозначим через $x^{(k)}$ текущее базисное решение с базисными компонентами $x^{(k)}_{B(i)} = b'_i,\ i = 1, ..., m$. Пусть $-c'_p < 0$, т.е. не выполнено условие оптимальности. Будем вводить в базис переменную x_p (или y_p, если ранее была сделана замена $x_p = u_p - y_p$). Как обычно, переход к базисному решению $x^{(k+1)}$ осуществляется увеличением компоненты x_p от нуля. При этом реберное решение имеет вид

$$\tilde{x} = (..., 0, ..., x_p, ..., \underset{B(i)}{b'_i - x_p a'_{ip}}, ..., 0, ...).$$

Увеличиваем x_p до тех пор, пока решение $\tilde{x}$ не перестанет быть допустимым в задаче (9.1). Чтобы компоненты решения $\tilde{x}$ были неотрицательны, необходимо выполнение неравенства

$$x_p \leqslant \theta = \min_{i:\ a'_{ip} > 0} \frac{b'_i}{a'_{ip}} = \frac{b'_l}{a'_{lp}}.$$

Еще нужно следить за выполнением неравенств вида

$$b'_i - x_p a'_{ip} \leqslant u_{B(i)}, \text{ где } B(i) \leqslant n.$$

Последнее неравенство может быть нарушено лишь при $a'_{ip} < 0$. Отсюда получаем, что

$$x_p \leqslant \theta_1 = \min_{i:\ a'_{ip}<0,\ B(i)\leqslant n} \frac{u_{B(i)} - b'_i}{-a'_{ip}} = \frac{u_{B(l)} - b'_l}{-a'_{lp}}.$$

Если $a'_{ip} \geqslant 0$ для всех таких i, что $B(i) \leqslant n$, то условно можно положить $\theta_1 = +\infty$. Итак, x_p не может превосходить каждую из трех величин: u_p, θ, θ_1, т.е. $x_p \leqslant \min[u_p, \theta, \theta_1]$.

Рассмотрим три возможных случая.

1) $\theta \leqslant \min[u_p, \theta_1]$.

Тогда компоненту x_p увеличиваем до величины $\theta = b'_l/a'_{lp}$, где l – номер ведущей строки. Делаем обычную операцию замещения, вводя в базис переменную x_p (или y_p) и выводя из него переменную $x_{B(l)}$.

2) $\theta_1 \leqslant \min[u_p, \theta]$.

В этом случае компоненту x_p увеличиваем до величины

$$\theta_1 = \frac{u_{B(l)} - b'_l}{-a'_{lp}}.$$

При этом $B(l)$-я компонента решения $\tilde{x}$ примет значение $u_{B(l)}$. Сделаем замену переменной $x_{B(l)} = u_{B(l)} - y_{B(l)}$. Поскольку в решении $\tilde{x}$ $x_{B(l)} = u_{B(l)}$, то $y_{B(l)} = 0$. Следовательно, переменная $y_{B(l)}$ будет выведена из базиса. Итак, после замены переменной необходимо сделать операцию замещения с ведущим элементом a'_{lp}. Нетрудно проверить, что в результате этих преобразований в решении $x^{(k+1)}$ базисная компонента x_p станет равной θ_1. Действительно, при замене переменной $x_{B(l)} = u_{B(l)} - y_{B(l)}$ из последнего столбца симплекс-таблицы вычитается базисный столбец e_l, умноженный на $u_{B(l)}$. В результате в последнем столбце изменится только один l-й элемент, который станет равным $b'_l - u_{B(l)}$. После операции замещения он станет равным $(b'_l - u_{B(l)})/a_{lp} = \theta_1$.

3) $u_p \leqslant \min[\theta, \theta_1]$.

Тогда в решении $\tilde{x}$ положим $x_p = u_p$ и сделаем замену переменной $x_p = u_p - y_p$. В результате $y_p = 0$. В этом случае базис остается прежним, но таблицу необходимо преобразовать в соответствии с заменой переменной.

После завершения преобразований проверяется условие оптимальности, т.е. неотрицательность нулевой строки симплекс-таблицы. Если оно выполнено, то решение $x^{(k+1)}$ оптимально. В противном случае итерации по алгоритму продолжаются.

Отметим, что если в исходной задаче (9.1) двусторонние ограничения накладываются только на часть переменных x_j, то проведенные рассуждения соответствующим образом модифицируются.

Пример 9.1. Рассмотрим задачу ЛП

$$Z = 2x_1 + x_2 + 2x_3 \quad \rightarrow \quad max$$

$$4x_1 + x_2 = 12, \quad -2x_1 + x_3 = 4,$$

$$0 \leqslant x_1 \leqslant 4, \quad 0 \leqslant x_2 \leqslant 15, \quad 0 \leqslant x_3 \leqslant 6.$$

Начальная симплекс-таблица имеет вид

$$\begin{array}{c|ccc|c} Z & -2 & -1 & -2 & 0 \\ \hline x_2 & 4 & 1 & 0 & 12 \\ x_3 & -2 & 0 & 1 & 4 \end{array}.$$

Обеспечим выполнение условия б) свойства (α) :

$$\begin{array}{c|ccc|c} Z & -2 & 0 & 0 & 20 \\ \hline x_2 & \underline{4} & 1 & 0 & 12 \\ x_3 & -2 & 0 & 1 & 4 \end{array}.$$

Здесь

$$p = 1, \ \ u_1 = 4, \ \ \theta = \frac{12}{4} = 3, \ \ \theta_1 = \frac{u_{B(2)} - b_2}{-a_{21}} = \frac{6-4}{2} = 1.$$

Поскольку $\theta_1 < \min[\theta, u_1]$, то нужно сделать замену переменной $x_3 = 6 - y_3$ и произвести операцию замещения с ведущим элементом a'_{21} :

$$\begin{array}{c|ccc|c} Z & -2 & 0 & 0 & 20 \\ \hline x_2 & 4 & 1 & 0 & 12 \\ y_3 & \underline{-2} & 0 & -1 & -2 \end{array}, \qquad \begin{array}{c|ccc|c} Z & 0 & 0 & 1 & 22 \\ \hline x_2 & 0 & 1 & -2 & 8 \\ x_1 & 1 & 0 & 1/2 & 1 \end{array}.$$

Отсюда находим, что $(x_1^*, x_2^*, y_3^*) = (1, 8, 0)$ – оптимальное решение и $x_3^* = 6$.

§10. Транспортная задача

Многие задачи ЛП формулируются как *транспортные* или сводятся к последним. Определим транспортную задачу. Имеется m пунктов производства некоторого продукта. В i-м пункте он выпускается в количестве $s_i, \ i = 1, ..., m$. Продукт может быть как штучным (в этом случае величины s_i – целые), так и бесконечно-делимым (например, водные ресурсы). Имеется n пунктов потребления продукта с потребностями $d_j, \ j = 1, ..., n$. Будем считать выполненным условие *баланса*

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j.$$

Пусть некоторая транспортная компания осуществляет перевозки продукта из пунктов производства в пункты потребления. Обозначим через c_{ij} стоимость перевозки единицы продукции из i-го пункта производства в j-й пункт потребления. Обычно стоимость перевозки пропорциональна расстоянию между пунктами.

План перевозок можно задать матрицей $x = (x_{ij})_{m \times n}$, согласно которому из i-го пункта в j-й перевозится продукт в количестве x_{ij}. Возникает следующая задача минимизации стоимости перевозок:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \quad \to \quad min$$

$$\sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, ..., m, \qquad \sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1, ..., n, \tag{10.1}$$

$$x_{ij} \geqslant 0, \quad i = 1, ..., m, \quad j = 1, ..., n,$$

которую будем называть стандартной транспортной задачей.

Часто возникают транспортные задачи, в которых условие баланса нарушено. В условиях дефицита продукта выполнено неравенство

$$\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j.$$

В этом случае также можно минимизировать стоимость перевозок, но некоторые пункты потребления продукт получат не полностью. Сведем эту задачу к стандартной. Введем $(m+1)$-й фиктивный пункт производства с величиной выпуска

$$s_{m+1} = \sum_{j=1}^{n} d_j - \sum_{i=1}^{m} s_i.$$

При этом следует положить $c_{m+1,j} = 0$, $j = 1, ..., n$, и рассмотреть задачу (10.1) с $m+1$ пунктами производства. Пусть x^* – оптимальное решение этой задачи. Если $x^*_{m+1,j} > 0$, то в j-й пункт потребления будет недопоставлена продукция в количестве $x^*_{m+1,j}$. Если необходимо, чтобы в j_1-й пункт была полная поставка, то следует положить $c_{m+1,j_1} = M$, где M – достаточно большое положительное число. Можно рассмотреть и такой вариант постановки задачи, когда компания платит штраф в c_j единиц за каждую недопоставленную в j-й пункт единицу продукции. Тогда следует положить $c_{m+1,j} = c_j$. Может возникнуть задача, когда перевозка из i-го пункта в j-й невозможна (например, при поставке воды из водоемов в города). Здесь также следует применить «метод большого M» и положить $c_{ij} = M$.

Предположим, что имеется избыток продукта, т.е.

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j.$$

Для сведения задачи к стандартной форме (10.1) введем фиктивный $(n+1)$-й пункт потребления и положим

$$d_{n+1} = \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j, \quad c_{i,n+1} = 0, \; i = 1, ..., m.$$

Если x^* – оптимальное решение этой задачи и $x^*_{i,n+1} > 0$, то в i-м пункте производства не будет взят продукт в количестве $x^*_{i,n+1}$. Как и в случае с дефицитом продукта, здесь возможны другие способы определения величин $c_{i,n+1}$, $i = 1, ..., m$.

Приведем два примера задач ЛП, которые в первоначальной постановке не являются транспортными, но по сути являются таковыми.

Задача о назначении. Завод имеет n цехов и закупил m станков. В каждом цехе можно установить не более одного станка. Пусть c_{ij} – стоимость установки i-го станка в j-м цехе. Требуется разместить станки по цехам с наименьшими затратами по их установке. Эту задачу можно рассматривать как транспортную с $d_j = s_i = 1$ при всех i, j. Если $m < n$, то возникает задача с дефицитом продукта.

Транспортная задача, как мы увидим, обладает следующим свойством. Если все величины s_i и d_j – целые числа, то существует оптимальное решение с целочисленными компонентами. Поэтому компоненты такого оптимального решения задачи о назначении равны нулю или единице.

Перейдем к изложению симплекс-метода решения стандартной транспортной задачи (10.1). Начнем с построения начального базисного решения. Прежде всего, заметим, что в задаче (10.1) mn переменных и $m+n$ ограничений-равенств. При этом сумма первых m уравнений равна сумме n последних. Следовательно, уравнения линейно зависимы.

Приведем два конкретных метода построения начального БДР.

1. Метод «северо-западного угла».

Берем $i_1 = 1$, $j_1 = 1$, т.е. элемент, стоящий в «северо-западном углу» матрицы C. После определения компоненты $x_{11}^{(1)}$ и вычеркивания первой строки или первого столбца в получившейся матрице берем элемент в «северо-западном углу» и т.д. Обычно этот метод дает неудачное начальное приближение, поскольку не учитывает стоимости c_{ij}.

2. Метод минимального элемента.

Выбираем пару (i_1, j_1) из условия $c_{i_1 j_1} = \min\limits_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} c_{ij}$. После вычеркивания в матрице C i_1-й строки или j_1-го столбца в получившейся матрице выбираем элемент (i_2, j_2) с минимальной стоимостью и т.д.

Пример 10.1. Рассмотрим транспортную задачу, заданную следующей таблицей:

					s_i
	$5\ \boxed{5}$	$\boxed{7}$	$\boxed{6}$	$\boxed{4}$	5
	$2\ \boxed{3}$	$0\ \boxed{4}$	$\boxed{3}$	$\boxed{2}$	2
	$\boxed{4}$	$1\ \boxed{3}$	$1\ \boxed{6}$	$1\ \boxed{5}$	3
d_j	7	1	1	1	

В правых верхних углах клеток указаны стоимости перевозок. В центрах клеток располагаются базисные компоненты начального решения

$$x^{(1)} = \begin{pmatrix} \underline{5} & 0 & 0 & 0 \\ \underline{2} & \underline{0} & 0 & 0 \\ 0 & \underline{1} & \underline{1} & \underline{1} \end{pmatrix}, \quad Z(x^{(1)}) = 45,$$

полученного по методу северо-западного угла. В матрице $x^{(1)}$ базисные элементы подчеркнуты. Если использовать метод минимального элемента, то

$$x^{(1)} = \begin{pmatrix} \underline{4} & 0 & \underline{1} & 0 \\ \underline{1} & 0 & 0 & \underline{1} \\ \underline{2} & \underline{1} & 0 & 0 \end{pmatrix}, \quad Z(x^{(1)}) = 42.$$

Нам потребуется некоторые свойства базисных компонент произвольного базисного решения $x^{(1)}$. Пусть $J^{(1)} = \{(i_t, j_t), \quad t = 1, ..., m+n-1\}$ – множество базисных пар, соответствующее $x^{(1)}$.

Лемма 10.1. Пусть $x^{(1)}$ – произвольное БДР с множеством базисных пар $J^{(1)}$. Тогда справедливы следующие утверждения:

1) В каждой строке и в каждом столбце матрицы C имеется хотя бы одна базисная пара.

2) Любая базисная пара не может быть единственной одновременно в своей строке и в своем столбце матрицы C.

3) Найдется строка (или столбец) матрицы C, содержащая (содержащий) единственную базисную пару.

Введем следующие бинарные отношения R_1 и R_2 на множестве пар (i, j) :

$$(i,j)R_1(i',j') \;\Leftrightarrow\; i = i', \qquad (i,j)R_2(i',j') \;\Leftrightarrow\; j = j'.$$

Очевидно, что отношения R_1 и R_2 транзитивны.

Будем рассматривать цепочки вида

$$(i_s, j_s)R_1(i_t, j_t)R_2(i_r, j_r)R_1....(i_l, j_l),$$

в которых отношения R_1 и R_2 чередуются и связывают базисные пары $(i_s, j_s), (i_t, j_t)$, $(i_r, j_r), ..., (i_l, j_l)$. Например, в базисном решении

$$x^{(1)} = \begin{pmatrix} \underline{4} & 0 & \underline{1} & 0 \\ \underline{1} & 0 & 0 & \underline{1} \\ \underline{2} & \underline{1} & 0 & 0 \end{pmatrix}$$

цепочка $(2,4)R_1(2,1)R_2(3,1)R_1(3,2)$ связывает базисные пары (2,4) и (3,2).

Теорема 10.1. Любые две базисные пары (i_s, j_s) и (i_t, j_t), отвечающие базисным компонентам начального базисного решения, связывает единственная цепочка.

Следствие. Если к базисным парам из $J^{(1)}$ добавить небазисную пару (i, j), то существует единственный цикл вида

$$(i,j)R_1(i,j_l)R_2...R_1(i_r,j)R_2(i,j),$$

где $(i, j_l)R_2...R_1(i_r, j)$ – цепочка, содержащая базисные пары.

Теперь перейдем непосредственно к методу решения транспортной задачи (10.1). В литературе он называется *методом потенциалов.* Задачу (10.1) можно записать как задачу максимизации функции

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} (-c_{ij}) x_{ij}.$$

Начнем симплекс-алгоритм с начального БДР $x^{(1)}$. Мы воспользуемся конкретной структурой матрицы ограничений. Заметим, что начальная симплекс-таблица имеет вид

$$\begin{array}{c|c|c} Z & c_{ij} & 0 \\ \hline x_{J^{(1)}} & A & b \end{array},$$

где $J^{(1)}$ – множество базисных пар решения $x^{(1)}$. Добьемся выполнения условия б) свойства (α) – элементы нулевой строки симплекс-таблицы, соответствующие базисным столбцам, должны быть нулевыми. Составим линейную комбинацию строк симплекс-таблицы, умножая каждую i-ю строку на u_i и каждую $(m+j)$-ю строку на v_j. Полученную комбинацию вычтем из нулевой строки, которая в результате преобразуется к виду

$$\begin{array}{c|c|c} Z & c_{ij} - u_i - v_j & -\sum\limits_{i=1}^{m} s_i u_i - \sum\limits_{j=1}^{n} d_j v_j \\ \hline & & \end{array}.$$

Для базисных пар $(i_t, j_t) \in J^{(1)}$ рассмотрим систему уравнений

$$u_{i_t} + v_{j_t} = c_{i_t j_t}, \;\; t = 1, ..., n+m-1, \tag{10.3}$$

и возьмем величины $u_1, ...u_m,\ v_1, ..., v_n$, удовлетворяющие этой системе. Тогда условие б) свойства (α) будет выполнено. Заметим, что система (10.3) содержит $m+n-1$ уравнений относительно $m+n$ неизвестных. Следовательно, эта система имеет бесконечно много решений. Как показывает результат упражнения 10.3, можно использовать любое из этих решений. Чтобы выделить единственное решение, положим $u_{\tilde{i}} = 0$, где $(\tilde{i}, \tilde{j})$ – базисная пара, имеющая наименьшую стоимость. При этом $v_{\tilde{j}} = c_{\tilde{i}\tilde{j}}$. Покажем, что и значения остальных переменных u_i, v_j определяются однозначно. Возьмем любую базисную пару $(i, j) \neq (\tilde{i}, \tilde{j})$. По теореме 10.1 пары $(\tilde{i}, \tilde{j})$ и (i, j) можно соединить единственной цепочкой

$$(\tilde{i}, \tilde{j}) R_1 (\tilde{i}, j_l) R_2 (i_k, j_l) .... R_1 (i_r, j) R_2 (i, j),$$

двигаясь вдоль которой можно найти $v_{\tilde{j}},\ u_{j_l}, ..., v_j$ и u_i. Поскольку в каждой строке и столбце найдется хотя бы одна базисная пара, то все переменные u_i и v_j будут однозначно определены.

Если для всех небазисных пар выполнено неравенство $c_{ij} - u_i - v_j \geqslant 0$, то начальное базисное решение оптимально. Пусть найдется пара (i, j), для которой выполнено неравенство $c_{ij} - u_i - v_j < 0$. Тогда можно ввести в базис переменную x_{ij}. Найдем переменную, которая выводится из базиса. Рассмотрим цикл, определенный в следствии теоремы 10.1

$$(i, j) R_1 (i, j_l) R_2 (i_k, j_l) .... R_1 (i_r, j) R_2 (i, j).$$

Построим новое базисное решение $x^{(2)}$. Для этого будем увеличивать компоненту $x_{ij} \geqslant 0$ и сохраним при этом все ограничения задачи (10.1). Для этого достаточно на величину x_{ij} уменьшить компоненту $x^{(1)}_{ij_l}$, увеличить на x_{ij} компоненту $x^{(1)}_{i_k j_l}$ и так далее, двигаясь вдоль цикла. При этом стоимость перевозок изменится на величину

$$x_{ij}(c_{ij} - c_{ij_l} + c_{i_k j_l} - ... - c_{i_r j}) =$$

$$= x_{ij}(c_{ij} - (u_i + v_{j_l}) + (u_{i_k} + v_{j_l}) - ... - (u_{i_r} + v_j)) = x_{ij}(c_{ij} - u_i - v_j),$$

т.е. уменьшится. Ясно, что x_{ij} можно увеличивать до величины

$$x^{(1)}_{i^*j^*} = \min x^{(1)}_{i_t j_t},$$

где минимум берется по базисным парам цикла, стоящим на четных местах. Покажем, что при $x_{ij} = x^{(1)}_{i^*j^*}$ мы получим новое базисное решение $x^{(2)}$. При этом переменная $x_{i^*j^*}$ будет выведена из базиса, а переменная x_{ij} – введена в него. Действительно, в разложении вектора A_{ij} по базисным столбцам

$$A_{ij} = A_{ij_l} - A_{i_k j_l} + ... + A_{i_r j}$$

вектор $A_{i^*j^*}$ входит с коэффициентом 1. Поэтому, если в системе базисных столбцов $A_{i_t j_t}$, $t = 1, ..., m + n - 1$, столбец $A_{i^*j^*}$ заменить столбцом A_{ij}, то получим снова линейно независимую систему.

Для продолжения симплекс-алгоритма необходимо вновь решить систему (10.3) для множества базисных пар

$$J^{(2)} = (J^{(1)} \backslash \{(i^*, j^*)\}) \cup \{(i, j)\}$$

и проверить условие оптимальности. Отметим, что величина $x^{(1)}_{i^*j^*}$ может быть равна нулю. Тогда $x^{(1)} = x^{(2)}$ и происходит только смена базиса.

Следствие. Если величины s_i, d_j – целые, то оптимальное базисное решение, полученное в результате работы алгоритма, будет также целочисленным. Действительно, начальное базисное решение – целочисленное. Далее целочисленность базисного решения будет поддерживаться, поскольку в алгоритме используются только операции сложения и вычитания.

Решим методом потенциалов пример 10.1 с таблицей

$u_i \backslash v_j$	3	2	4	2	s_i
2	$\boxed{5}$ 4	$\boxed{7}$	$\boxed{6}$ 1	$\boxed{4}$	5
0	$\boxed{3}$ 1	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$ 1	2
1	$\boxed{4}$ 2	$\boxed{3}$ 1	$\boxed{6}$	$\boxed{5}$	3
d_j	7	1	1	1	

В центрах клеток таблицы указаны базисные компоненты начального БДР $x^{(1)}$, полученного по методу минимальной стоимости. Этим компонентам отвечает система уравнений

$$u_1 + v_1 = 5, \qquad u_1 + v_3 = 6,$$

$$u_2 + v_1 = 3, \qquad u_2 + v_4 = 2, \qquad u_2 = 0,$$

$$u_3 + v_1 = 4, \qquad u_3 + v_2 = 3.$$

Решение системы указано слева и сверху от таблицы. Условие оптимальности не выполнено для пары (2,3): $c_{23} - u_2 - v_3 = -1 < 0$. Находим цикл, соединяющий пару (2,3) с базисными парами

$$(2,3)R_1(2,1)R_2(1,1)R_1(1,3)R_2(2,3).$$

На четных местах этого цикла компоненты решения $x^{(1)}$ равны 1. Поэтому переменную x_{23} можно увеличить на 1. В результате получим таблицу

u_i \ v_j	3	2	3	2	s_i
2	5 $\boxed{5}$	$\boxed{7}$	$\boxed{6}$	$\boxed{4}$	5
0	0 $\boxed{3}$	$\boxed{4}$	1 $\boxed{3}$	1 $\boxed{2}$	2
1	2 $\boxed{4}$	1 $\boxed{3}$	$\boxed{6}$	$\boxed{5}$	3
d_j	7	1	1	1	

Здесь переменная x_{13} выведена из базиса, а переменная x_{23} введена в него. Решаем систему уравнений

$$u_1 + v_1 = 5, \qquad u_2 + v_3 = 3,$$
$$u_2 + v_1 = 3, \qquad u_2 + v_4 = 2, \qquad u_2 = 0,$$
$$u_3 + v_1 = 4, \qquad u_3 + v_2 = 3.$$

Теперь условие оптимальности выполнено и $x^{(2)}$ – оптимальное решение задачи стоимости $Z(x^{(2)}) = 41$.

§11. Метод ветвей и границ

Сначала изложим общую схему метода, а потом его конкретную реализацию для задачи целочисленного ЛП

$$Z = \sum_{j=1}^{n} c_j x_j \quad \to \quad max$$
$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \; i = 1, ..., m, \;\; x_j \geqslant 0, \; x_j \in \mathbb{Z}, \;\; j = 1, ..., n, \tag{11.0}$$

где $\mathbb{Z}$ – множество целых чисел.

Пусть целевая функция $Z(x)$ определена на конечном множестве решений X_0. Рассмотрим задачу:

$$\max_{x \in X_0} Z(x) = Z(x^0). \tag{11.1}$$

Обычно имеется естественный способ «погружения» множества X_0 в более «широкое» множество $X \supset X_0$.

Предположим, что мы умеем разбивать (ветвить) множество X на такие подмножества A, для которых подзадача

$$\max_{x \in A} Z(x) = Z(x_A) = Z_A$$

решается сравнительно несложно. Если оптимальное решение x_A подзадачи является допустимым (т.е. $x_A \in X_0$), то величина Z_A представляет собой оценку (границу) снизу для искомого максимума $Z(x^0)$.

Рассмотрим метод ветвей и границ для задачи (11.1) подробнее. Пусть N – текущая оценка снизу для максимума $Z(x^0)$ задачи (11.1). Эта оценка либо равна $-\infty$, либо достигается на некотором текущем допустимом решении. Вначале полагаем $N = -\infty$ и решаем задачу

$$\max_{x \in X} Z(x) = Z(x^*) = Z_X.$$

Если $x^* \in X_0$, то x^* – оптимальное решение задачи (11.1), поскольку $X_0 \subset X$.

Предположим, что $x^* \notin X_0$. В этом случае множество X разбиваем на подмножества $X_1, ..., X_m$, последовательно решаем подзадачи и находим величины $Z_{X_i} = Z(x_{X_i}),\ i = 1, ..., m$. При этом по следующим правилам производим отбрасывание подмножеств X_i :

1) $N \geqslant Z_{X_i}$. В этом случае N – конечная величина, которая по построению достигается на текущем допустимом решении задачи (11.1). Поэтому все решения из множества X_i имеют не большие значения целевой функции и могут быть отброшены.

2) $N < Z_{X_i}$ и $x_{X_i} \in X_0$, т.е. оптимальное решение подзадачи допустимо в задаче (11.1). Тогда полагаем $N = Z_{X_i}$ и запоминаем x_{X_i} как новое текущее решение. В этом случае оценка N улучшается.

3) Множество X_i не содержит допустимых решений.

После просмотра всех подмножеств X_i отбираем те из них, которые не были отброшены. Выбираем среди них подмножество с наибольшим значением Z_{X_i} и разбиваем его на подмножества $X_{ij},\ j = 1, ..., n$. Решаем подзадачи с подмножествами X_{ij}, используя правила отбрасывания подмножеств и т.д. Алгоритм завершает свою работу, когда все подмножества будут отброшены.

Заметим, что в каждой конкретной реализации метода ветвей и границ необходимо доказывать его сходимость за конечное число шагов. Покажем, что при неудачном способе ветвления задачи алгоритм может привести к бесконечному числу шагов. Предположим, что в задаче целочисленного ЛП имеется единственная целочисленная точка x^0, внутренняя для множества допустимых решений X непрерывной задачи. Пусть процесс разбиения на подмножества ведется таким образом, что точка x^0 всегда остается внутренней для содержащего ее множества разбиения. Оптимальные решения подзадач, полученные симплекс-методом, принадлежат границам подмножеств разбиения. Поэтому в этом примере метод ветвей и границ не сходится за конечное число шагов.

Продемонстрируем метод ветвей и границ на примере решения задачи о назначении.

Пример 11.1. Пусть $\alpha, \beta, \gamma, \delta$ – рабочие, которых необходимо распределить по четырем работам с номерами 1,2,3,4. Здесь множество X_0 – состоит из 24 перестановок символов $\alpha, \beta, \gamma, \delta$. Например, $x^1 = (\delta, \gamma, \alpha, \beta)$ означает, что рабочий δ выполняет первую работу, рабочий γ – вторую и т.д. В следующей матрице

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} \alpha \\ \beta \\ \gamma \\ \delta \end{array} & \begin{pmatrix} 9 & 1 & 5 & 6 \\ 7 & 2 & 8 & 8 \\ 6 & 8 & 0 & 7 \\ 1 & 7 & 3 & 2 \end{pmatrix} \end{array} \tag{11.2}$$

содержатся величины эффективности использования рабочих на каждой работе. Определим функцию $Z(x)$ как суммарную величину эффективности распределения рабочих по работам. Например, $Z(x^1) = 1 + 8 + 5 + 8 = 22$.

Рассмотрим множество X, состоящее из векторов x, по которым допускается распределение рабочего более, чем на одну работу. Множество X включает X_0 и содержит 256 решений.

Нетрудно видеть, что

$$\max_{x \in X} Z(x) = Z(x^*) = 9 + 8 + 8 + 8 = 33, \;\; x^* = (\alpha, \gamma, \beta, \beta) \notin X_0.$$

Переходим к разбиению задачи на подзадачи. Пусть X_α – множество таких x, для которых $x_1 = \alpha, \; x_j \neq \alpha, \; j = 2, 3, 4$. Иначе, при $x \in X_\alpha$ только рабочий α назначается на первую работу. Аналогично определяются множества $X_\beta, X_\gamma, X_\delta, X_{\alpha\beta}$ и т.п. Тогда

$$X = X_\alpha \cup X_\beta \cup X_\gamma \cup X_\delta \cup \tilde{X},$$

где $\tilde{X}$ – состоит из векторов x, у которых первая компонента совпадает с какой-нибудь другой. Поскольку множество $\tilde{X}$ не содержит допустимых решений, его можно сразу отбросить.

Алгоритм удобно проиллюстрировать с помощью дерева ветвления задачи. Вершина дерева будет соответствовать некоторому подмножеству A множества X. Она изображается кружком, внутри которого указано значение Z_A. Если $x_A \in X_0$, то значение Z_A будем подчеркивать. Из корневой вершины дерева выходят четыре ребра, отмеченные символами $\alpha, \beta, \gamma, \delta$. Они соответствуют выборам подзадач с подмножествами $X_\alpha, X_\beta, X_\gamma, X_\delta$ множества X. Следующее за ребром α ребро с символом β соответствует выбору подзадачи с подмножеством $X_{\alpha\beta}$ и т.д. На рис. 11.1 представлены результаты вычислений.

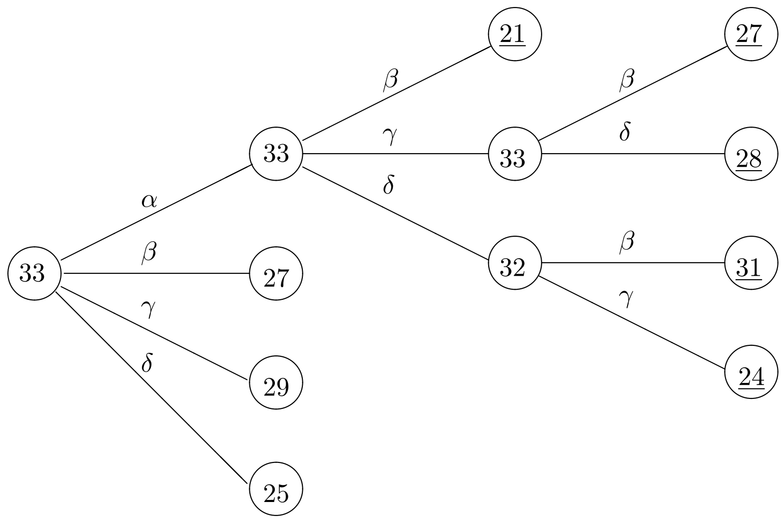

Рис. 11.1.

Здесь $x^0 = (\alpha, \delta, \beta, \gamma)$ – оптимальное решение и $Z(x^0) = 31$.

Применим метод ветвей и границ для решения задачи целочисленного линейного программирования (11.0). Будем считать, что множество X допустимых решений соответствующей непрерывной задачи (1.1) является многогранником. Обозначим через $\lfloor a \rfloor$ целую часть числа a. Пусть x^* – оптимальное решение непрерывной задачи (1.1). Если x^* содержит целые компоненты, то $x^* \in X_0$ и x^* – оптимальное решение задачи ((11.0)). Пусть решение x^* содержит нецелые компоненты и x^*_s – какая-нибудь из них. Определим три множества

$$X_1 = \{x \in X \mid x_s \leqslant \lfloor x^*_s \rfloor\}, \;\; X_2 = \{x \in X \mid x_s \geqslant \lfloor x^*_s \rfloor + 1\},$$

$$\tilde{X} = \{x \in X \mid \lfloor x^*_s \rfloor < x_s < \lfloor x^*_s \rfloor + 1\}.$$

Поскольку $\tilde{X}$ не содержит допустимых решений задачи ((11.0)), оно сразу же может быть отброшено. В соответствии с методом ветвей и границ находим оптимальные решения подзадач x^1, x^2 : $Z_{X_1} = Z(x^1)$, $Z_{X_2} = Z(x^2)$. Одно из множеств X_1 или X_2 может оказаться пустым, тогда его отбрасываем. Если $x^1 \in X_0$, то полагаем $N = Z_{X_1}$, решение x^1 запоминаем, а множество X_1 отбрасываем. Аналогичную проверку делаем для x^2. Предположим, что оба множества X_1 и X_2 не были отброшены и что $Z_{X_1} \geqslant Z_{X_2}$. Тогда берем множество X_1 для дальнейшего ветвления. Для нецелой компоненты x^1_k определим два множества

$$X_3 = \{x \in X_1 \mid x_k \leqslant \lfloor x^1_k \rfloor\}, \;\; X_4 = \{x \in X_1 \mid x_k \geqslant \lfloor x^1_k \rfloor + 1\}.$$

Решаем соответствующие подзадачи и т.д. Алгоритм заканчивает работу за конечное число шагов. Действительно, при каждом ветвлении задачи по какой-нибудь оси x_i отбрасывается интервал (множество вида $\tilde{X}$.) Поскольку X – многогранник, таких интервалов конечное число и отсюда вытекает конечность алгоритма.

Рассмотрим конкретный числовой пример:

$$Z = 8x_1 + 8x_2 + 5x_3 \quad \to \quad max$$

$$6x_1 + 8x_2 + 4x_3 \leqslant 1360, \;\; 6x_1 + 8x_2 \geqslant 1250,$$

$$0 \leqslant x_1 \leqslant 100, \; 0 \leqslant x_2 \leqslant 160, \; 0 \leqslant x_3 \leqslant 120, \;\; x_j \in \mathbb{Z}, \;\; j = 1, 2, 3.$$

Решим соответствующую «непрерывную» задачу. Введем для первого неравенства слабую переменную x_4, а для второго $-$ x_5.

Выпишем начальную симплекс-таблицу

Z	-8	-8	-5	0	0	0
x_4	6	8	4	1	0	1360
x_5	-6	$\underline{-8}$	0	0	1	-1250

.

Преобразуем ее с целью получения начального базисного решения. Для этого сделаем операцию замещения с ведущим элементом $a_{22} = -8$:

Z	-2	0	-5	0	-1	1250
x_4	0	0	4	1	1	110
x_2	$3/4$	1	0	0	$-1/8$	$625/4$

.

Возьмем ведущим первый столбец. Тогда $\theta = 625/3 > u_1 = 100$ и, следовательно, необходимо сделать замену $x_1 = 100 - y_1$:

Z	2	0	-5	0	-1	1450
x_4	0	0	4	1	1	110
x_2	$-3/4$	1	0	0	$-1/8$	$325/4$

.

Возьмем ведущим третий столбец. Тогда $\theta = 110/4 < u_3 = 120$ и первая строка $-$ ведущая. Делаем обычное преобразование таблицы:

Z	2	0	0	$5/4$	$1/4$	$3175/2$
x_3	0	0	1	$1/4$	$1/4$	$55/2$
x_2	$-3/4$	1	0	0	$-1/8$	$325/4$

.

Находим оптимальное решение «непрерывной» задачи: $y_1^* = 0$, $x_2^* = 325/4$, $x_3^* = 55/2$. Окончательно $x^* = (100, 325/4, 55/2)$.

Возьмем нецелую компоненту x_2^* и рассмотрим множества

$$X_1 = \{x \in X \mid x_2 \leqslant 81\}, \quad X_2 = \{x \in X \mid x_2 \geqslant 82\},$$

где X $-$ множество решений $x = (x_1, x_2, x_3)$, удовлетворяющих ограничениям задачи без требования целочисленности компонент.

Нетрудно видеть, что множество X_1 пусто. Действительно, для
$x \in X_1$

$$6x_1 + 8x_2 \leqslant 600 + 648 = 1248 < 1250$$

и второе неравенство из ограничений не выполнено.

Решим подзадачу $\max\limits_{x\in X_2} Z(x) = Z(x^0)$. От только что решенной задачи она отличается неравенством $82 \leqslant x_2 \leqslant 100$. Воспользуемся уже проведенными вычислениями. Сделаем замену $x_2 = \tilde{x}_2 + 82$. Тогда $0 \leqslant \tilde{x}_2 \leqslant 18$. Заключительная симплекс-таблица преобразуется к виду

Z	2	0	0	$5/4$	$1/4$	$3175/2$
x_3	0	0	1	$1/4$	$1/4$	$55/2$
x_2	$-3/4$	1	0	0	$-1/8$	$-3/4$

.

Итак, текущее базисное решение не является допустимым. Воспользуемся двойственным симплекс-методом. Вторая строка – ведущая, $\theta' = \min[8/3, 2] = 2$ и пятый столбец – ведущий. После преобразования получаем симплекс-таблицу

Z	$1/2$	2	0	$5/4$	0	1586
x_3	$-3/2$	2	1	$1/4$	0	26
x_5	6	-8	0	0	1	6

.

Отсюда $y_1^0 = 0,\ \tilde{x}_2^0 = 0,\ x_3^0 = 26$. Окончательно $x^0 = (100, 82, 26)$. Поскольку решение x^0 состоит из целых компонент, оно является оптимальным решением исходной задачи.

§12. Метод динамического программирования

Вначале рассмотрим общий *динамический процесс* принятия решений, протекающий за n шагов. Каждому шагу k соответствует множество *состояний* S_k. Если процесс находится в состоянии $s_k \in S_k$, то принимается решение о выборе *альтернативы* $x_k \in X_k(s_k)$. После этого процесс переходит в состояние $s_{k+1} = \sigma_k(s_k, x_k)$, отвечающее следующему, $(k+1)$-у шагу, которое вычисляется по функции $\sigma_k(s_k x_k)$. Для простоты будем считать, что на первом шаге множество S_1 содержит единственное состояние s_1. Процесс заканчивается в состоянии $s_{n+1} \in S_{n+1}$, где выбор альтернатив не производится. Для каждой пары s_k, x_k определена оценка $f_k(s_k, x_k)$ альтернативы x_k в состоянии s_k. *Наилучшая альтернатива* $x_k^*(s_k)$ определяется из условия

$$f_k^*(s_k) \stackrel{def}{=} \max_{x_k\in X_k(s_k)} f_k(s_k, x_k) = f_k(s_k, x_k^*(s_k)).$$

Часто вместо максимума здесь пишут минимум, если это соответствует смыслу оценок.

Функция $f_k^*(s_k)$ называется *функцией Беллмана.* В методе динамического программирования предполагается, что $f_k^*(s_k)$ является наилучшим результатом, который можно обеспечить при условии, что, начиная с k-го шага, применяются наилучшие альтернативы. Поэтому $f_k(s_k, x_k)$ – оценка альтернативы $x_k \in X_k(s_k)$ при условии, что начиная с $(k+1)$-го шага, применяются наилучшие альтернативы. Отсюда следует, что функция $f_k(s_k, x_k)$ должна быть задана в более конкретном виде

$$f_k(s_k, x_k) = g_k(x_k, f_{k+1}^*(s_{k+1})),$$

где $s_{k+1} = \sigma_k(s_k, x_k)$. Отсюда и из определения функции $f_k^*(s_k)$ получаем

$$f_k^*(s_k) = \max_{x_k\in X_k(s_k)} g_k(x_k, f_{k+1}^*(\sigma_k(s_k, x_k))).$$

Последнее соотношение носит название *уравнения Беллмана.* Функция $f^*_{n+1}(s_{n+1})$ в зависимости от смысла задачи полагается равной тождественно нулю или единице.

Уравнение Беллмана позволяет подсчитать функции $f^*_k(s_k)$ и $x^*_k(s_k)$, начиная с n-го и заканчивая первым шагом. Целью вычислений по методу динамического программирования является нахождение величины $f^*_1(s_1)$ и последовательности $x^*_1 = x^*_1(s_1)$, $s^*_2 = \sigma_1(s_1, x^*_1)$, $x^*_2 = x^*_2(s^*_2), \ldots$, $s^*_n = \sigma_{n-1}(s^*_{n-1}, x^*_{n-1})$, $x^*_n = x^*_n(s^*_n)$, содержащей оптимальные значения альтернатив x^*_k для всех шагов процесса.

Отметим, что когда найдена функция $f^*_k(s_k)$, функцию $f^*_{k+1}(s_{k+1})$ можно «забыть». Однако, все функции $x^*_k(s_k)$ необходимо помнить, что рассматривается как недостаток метода. Для его применения в какой-либо задаче необходимо конкретизировать элементы рассмотренного динамического процесса и пояснить смысл функций Беллмана.

Пример 12.1. На рис. 3.1 указано расположение 10 городов, связанных между собой дорогами (цифры рядом со стрелками – длины дорог, под квадратиками – номера городов). Предполагается, что по дорогам можно перемещаться только в направлениях, указанных стрелками. Требуется проехать из города 1 в город 10 по кратчайшему пути.

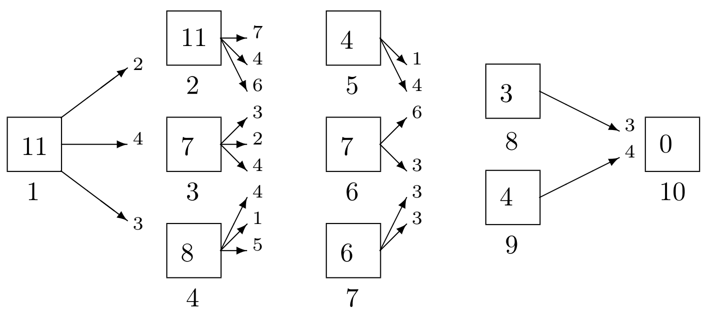

Рис. 3.1.

Здесь естественно считать, что процесс перемещения из города 1 в город 10 происходит в течение 4-х шагов. На первом шаге $S_1 = \{1\}$. На втором множество состояний равно $S_2 = \{2, 3, 4\}$ и т.д. На k-м шаге пусть x_k – дорога, ведущая из города s_k в следующий город $s_{k+1} = \sigma_k(s_k, x_k)$. Введем функцию оценки $f_k(s_k, x_k)$. Она равна длине кратчайшего пути из города s_k в город 10, если ехать сначала по дороге x_k. Тогда функция Беллмана

$$f^*_k(s_k) = \min_{x_k \in X_k(s_k)} f_k(s_k, x_k)$$

– длина кратчайшего пути из города s_k в город 10. Заметим, что

$$f_k(s_k, x_k) = d(x_k) + f^*_{k+1}(s_{k+1}),$$

где $d(x_k)$ – длина дороги x_k. На рисунке значения функций Беллмана указаны внутри квадратиков. В частности, $f^*_1(s_1) = 11$ – длина кратчайшего пути $1 \to 3 \to 5 \to 10$ из города 1 в город 10.

Рассмотрим задачу максимизации *сепарабельной* целевой функции

$$\begin{gathered} Z = \sum_{j=1}^{n} p_j(x_j) \quad \to \quad max \\ \sum_{j=1}^{n} a_j x_j \leqslant A, \quad x_j \geqslant 0, \ \ x_j \in \mathbb{Z}, \ \ j = 1, ..., n. \end{gathered} \tag{12.1}$$

Здесь константы A, a_j, $j = 1, ..., n$, положительны, а $p_j(x_j)$ – возрастающие функции целого аргумента.

Задачу (12.1) можно интерпретировать как задачу оптимального размещения рекламных объявлений. Пусть некоторая фирма проводит рекламную компанию. Имеется n пунктов возможного размещения рекламы (газеты, радио, телевидение и т.п.) Исследования показали, что если в j-м пункте объявление дается t раз, то ожидаемое число клиентов фирмы возрастает на величину $p_j(t)$. Пусть A – количество денег, выделенных на рекламу, а a_j – стоимость одного рекламного объявления в j-м пункте размещения.

Рассмотрим частный случай, когда $a_j = 1, \quad j = 1, ..., n$, а A – целое число. Тогда максимум в задаче (12.1), очевидно, достигается при выполнении равенства $\sum\limits_{j=1}^{n} x_j = A$. В результате получим задачу

$$\begin{gathered} Z = \sum_{j=1}^{n} p_j(x_j) \quad \to \quad max \\ \sum_{j=1}^{n} x_j = A, \qquad x_j \geqslant 0, \ \ x_j \in \mathbb{Z}, \ \ j = 1, ..., n. \end{gathered} \tag{12.2}$$

Задаче (12.2) можно дать другую интерпретацию. Страховая компания распределяет A своих агентов по n районам города. Если в j-м районе работает t агентов, то $p_j(t)$ – ожидаемое число заключенных договоров страхования. Необходимо так распределить агентов по районам, чтобы общее ожидаемое число договоров было наибольшим.

Применим метод динамического программирования к задаче (12.1). Будем считать, что значения переменных выбираются последовательно: сначала x_1, затем x_2 и т.д. Положим $s_k = A - \sum\limits_{j=1}^{k-1} a_j x_j$ – количество денег, оставшееся после размещения рекламы по первым $k-1$ пунктам. При этом

$$s_k \in S_k \subset [0, A], \ \ s_{k+1} = \sigma_k(s_k, x_k) = s_k - a_k x_k.$$

Отметим, что сумма

$$\sum_{j=1}^{k-1} a_j x_j$$

принимает лишь конечное число значений (x_j – целые). В задаче (12.2) $S_k = \{0, ..., A\}$. Далее,

$$x_k \in X_k(s_k) = \{0, 1, ..., \lfloor s_k / a_k \rfloor\}.$$

Определим множество

$$D_k(s_k) = \Big\{(x_k, ..., x_n) \;\Big|\; \sum_{j=k}^{n} a_j x_j \leqslant s_k,\ x_j \geqslant 0, \quad x_j \in \mathbb{Z},\ j = k, ..., n\Big\}$$

и функцию, соответствующую k-му шагу,

$$f_k^*(s_k) = \max_{(x_k,...,x_n) \in D_k(s_k)} \sum_{j=k}^{n} p_j(x_j). \tag{12.3}$$

Решая задачу (12.3), фирма наилучшим образом распределяет деньги в количестве s_k по пунктам с номерами $k, k+1, ..., n$.

Теперь определим функцию

$$f_k(s_k, x_k) = p_k(x_k) + f_{k+1}^*(s_k - a_k x_k),$$

оценивающую выбор x_k в состоянии s_k. Эта оценка основана на том, что после выбора x_k оставшаяся сумма $s_k - a_k x_k$ распределяется оптимально по пунктам с номерами $k + 1, ..., n$. Будем считать, что $f_{n+1}^*(s_{n+1}) \equiv 0$. Покажем, что функции $f_k^*(s_k)$ удовлетворяют уравнению Беллмана

$$f_k^*(s_k) = \max_{x_k \in X_k(s_k)} (p_k(x_k) + f_{k+1}^*(s_k - a_k x_k)),\ k = n, n-1, ..., 1.$$

При $k = n$

$$f_n^*(s_n) = \max_{x_n \in X_n(s_n)} p_n(x_n) = p_n(\lfloor s_n/a_n \rfloor),\ \ x_n^*(s_n) = \lfloor s_n/a_n \rfloor$$

и утверждение очевидно.

Пусть $k < n$. Возьмем такой вектор $(x_k^*, ..., x_n^*) \in D_k(s_k)$, что $f_k^*(s_k) = \sum\limits_{j=k}^{n} p_j(x_j^*)$. Тогда $x_k^* \in X_k(s_k)$ и $(x_{k+1}^*, ..., x_n^*) \in D_{k+1}(s_k - a_k x_k^*)$. Отсюда

$$f_k^*(s_k) \leqslant \max_{x_k \in X_k(s_k)} (p_k(x_k) + f_{k+1}^*(s_k - a_k x_k)).$$

Аналогично доказывается, что

$$f_k^*(s_k) \geqslant \max_{x_k \in X_k(s_k)} (p_k(x_k) + f_{k+1}^*(s_k - a_k x_k)).$$

Из уравнения Беллмана последовательно находим функции

$$f_k^*(s_k),\ \ x_k^*(s_k), \quad k = n-1, n-2, ..., 1.$$

Поскольку $s_1 = A$, то $s_2 = A - a_1 x_1^*(A)$, $s_3 = s_2 - a_2 x_2^*(s_2), ...$. В результате получаем $x^* = (x_1^*(A), x_2^*(s_2), ..., x_n^*(s_n))$ – оптимальное решение задачи (12.1).

Если в задаче (12.1) числа n и A велики, то при реализации метода динамического программирования сохранение функций $f_k^*(s_k)$ и $x_k^*(s_k)$ может потребовать значительный объем памяти. Этих сложностей меньше в задаче (12.2), где величина s_k при любом k

принимает значения $0, 1, ..., A$. Аналогичные упрощения возможны и в случае, когда все величины a_j целые.

Пример 12.2. Организация решила сделать рекламные объявления в газете, на телевидении и по радио. Стоимость одного объявления в газете составляет 220 условных единиц, на телевидении – 300 единиц и по радио – 250 единиц. Расходы на рекламу планируются в размере 1000 единиц. Пусть $p_j(t)$ – ожидаемое число новых клиентов после t объявлений в j-м пункте размещения рекламы, $j = 1, 2, 3$. Функции $p_j(t)$ заданы следующим образом:

t	$p_1(t)$	$p_2(t)$	$p_3(t)$
0	0	0	0
1	20	30	25
2	40	54	50
3	55	82	70
4	70	100	90

Требуется наиболее эффективно использовать деньги на рекламу. Получаем задачу (12.1) с $A = 1000,\ a_1 = 220,\ a_2 = 300,\ a_3 = 250$. Применим метод динамического программирования. Функции $f_3^*(s_3) = p_3(\lfloor s_3/250 \rfloor)$ и $x_3^*(s_3) = \lfloor s_3/250 \rfloor$ кусочно-постоянны. Выпишем их значения в таблицу

s_3	$f_3^*(s_3)$	$x_3^*(s_3)$
$0 \leqslant s_3 < 250$	0	0
$250 \leqslant s_3 < 500$	25	1
$500 \leqslant s_3 < 750$	50	2
$750 \leqslant s_3 < 1000$	70	3
$s_3 = 1000$	90	4

Теперь протабулируем функции

$$f_2(s_2, x_2) = p_2(x_2) + f_3^*(s_2 - 300x_2),$$

$$f_2^*(s_2) = \max_{x_2 \in X_2(s_2)} f_2(s_2, x_2) \quad \text{и} \quad x_2^*(s_2).$$

Заметим, что величина

$$s_2 = 1000 - 220x_1 \in S_2 = \{120, 340, 560, 780, 1000\}.$$

Выпишем таблицу

s_2	$f_2(s_2,0)$	$f_2(s_2,1)$	$f_2(s_2,2)$	$f_2(s_2,3)$	$f_2^*(s_2)$	$x_2^*(s_2)$
120	0				0	0
340	25	30			30	1
560	50	55			55	1
780	70	55	54		70	0
1000	90	80	79	82	90	0

Осталось найти

$$f_1(s_1, x_1) = p_1(x_1) + f_2^*(s_1 - 220x_1),$$

$$f_1^*(s_1) = \max_{x_1 \in X_1(s_1)} f_1(s_1, x_1), \quad x_1^*(s_1)$$

при одном значении $s_1 = 1000$. Получим таблицу

$f_1(s_1,0)$	$f_1(s_1,1)$	$f_1(s_1,2)$	$f_1(s_1,3)$	$f_1(s_1,4)$	$f_1^*(s_1)$	$x_1^*(s_1)$
90	90	95	85	70	95	2

Отсюда

$$x_1^* = x_1^*(s_1) = 2,\ s_2^* = s_1 - 2 \cdot 220 = 560, x_2^* = x_2^*(560) = 1,$$
$$s_3^* = 560 - 1 \cdot 300 = 260,\ x_3^* = x_3^*(260) = 1.$$

Таким образом, $x^* = (2, 1, 1)$ – оптимальное решение задачи. Организации следует сделать два объявления в газете и по одному по телевидению и по радио.

Рассмотрим еще одну задачу

$$\begin{gathered} Z = \prod_{j=1}^{n} p_j(x_j) \quad \to \quad max \\ \sum_{j=1}^{n} a_j(x_j) \leqslant A, \ \ 1 \leqslant x_j \leqslant m, \ \ x_j \in \mathbb{Z}, \quad j = 1, ..., n. \end{gathered} \tag{12.4}$$

Здесь $p_j(x_j), a_j(x_j)$ – возрастающие функции целого аргумента.

Задача (12.4) имеет следующую интерпретацию. Проектируется система, состоящая из n элементов. Каждый j-й элемент может быть дублирован. При этом число x_j параллельно соединенных элементов не может быть больше m. $p_j(x_j)$ – вероятность невыхода из строя j-го дублированного элемента, а $a_j(x_j)$ – его стоимость. Тогда Z – вероятность невыхода из строя всей системы (ее надежность). Пусть A – сумма денежных средств, предназначенных для создания системы. Требуется спроектировать систему, имеющую наибольшую надежность.

Применим метод динамического программирования. Естественно считать, что

$$s_k = A - \sum_{j=1}^{k-1} a_j(x_j)$$

– сумма средств, оставшаяся после выбора значений $x_1, ..., x_{k-1}$ на предыдущих шагах. При этом

$$s_k \in S_k \subset \Big[\sum_{j=k}^{n} a_j(1), \ \ A - \sum_{j=1}^{k-1} a_j(1) \Big].$$

Для $s_k \in S_k$ определим $x_k^0(s_k) = \max\{x_k \mid a_k(x_k) \leqslant s_k\}$. Тогда $x_k \in X_k(s_k) = \{1, ..., x_k^0(s_k)\}$. Положим

$$f_{n+1}^*(s_{k+1}) \equiv 1, \quad \sigma_k(s_k, x_k) = s_k - a_k(x_k),$$
$$f_k^*(s_k) = \max_{x_k \in X_k(s_k)} (p_k(x_k) f_{k+1}^*(s_k - a_k(x_k)))$$

– уравнение Беллмана. Далее заметим, что

$$f_n^*(s_n) = \max_{x_n \in X_n(s_n)} p_n(x_n) = p_n(x_n^0(s_n)), \quad x_n^*(s_n) = x_n^0(s_n).$$

Как и для случая сепарабельной функции, из уравнения Беллмана последовательно находим функции

$$f_k^*(s_k), \;\; x_k^*(s_k), \quad k = n-1, n-2, ..., 1.$$

Далее $s_1 = A,\; s_2 = A - a_1(x_1^*(A)),\; s_3 = s_2 - a_2(x_2^*(s_2)), ...$ В результате получаем $x^* = (x_1^*(A), x_2^*(s_2), ..., x_n^*(s_n))$ – оптимальное решение задачи (12.4).