

## Глава 12. ЭЛЕМЕНТЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Терминологическое замечание. Мы считаем необходимым отметить, что следует различать *программирование*, т.е. составление программ для ЭВМ, и *математическое программирование*, т.е. исследование и решение задач оптимизации (минимизации или максимизации) вещественного функционала, заданного на  $\mathbb{R}^n$  или на его части.

Частным случаем математического программирования является *линейное программирование* – оптимизация линейного функционала на части  $\mathbb{R}^n$ , заданной *линейными* ограничениями (равенствами или неравенствами).

Линейное программирование ведет свою историю от работы Л.В. Канторовича<sup>1</sup>, выполненной в 1938 году.

### 12.1. Одна содержательная задача

Описание проблемы, которой посвящена эта глава, мы начнем с простейшего примера.

Коммерсант, выехавший для закупки двух видов товара, имеет 18 денежных единиц (д.е.)<sup>2</sup>, его автомобиль может вместить 10 единиц массы (е.м.). Одна е.м. товара первого вида стоит 1 д.е., второго вида – 3 д.е. При продаже 1 е.м. товара первого вида коммерсант рассчитывает получить 0.5 д.е. прибыли, при продаже 1 е.м. товара второго вида – 0.75 д.е. Как распределить имеющиеся деньги и вместимость автомобиля, чтобы ожидаемая прибыль была максимальной?

Пусть  $x_1$  – закупаемое коммерсантом количество товара первого вида,  $x_2$  – второго вида. Эти переменные должны удовлетворять следующим очевидным неравенствам:

- 1)  $x_1 \geq 0, \quad x_2 \geq 0$  (количества товаров неотрицательны);
- 2)  $x_1 + 3x_2 \leq 18$  (затрачиваемая сумма не может превышать наличность);
- 3)  $x_1 + x_2 \leq 10$  (суммарная масса закупленных товаров не может превышать вместимость автомобиля).

На части  $\mathbb{R}^2$ , где выполнены все эти неравенства, требуется найти наибольшее значение линейного функционала

$$f(x) = 0.5x_1 + 0.75x_2.$$

Рассмотрим геометрическую интерпретацию сформулированной задачи. Множество, задаваемое одним линейным неравенством, представляет собой полуплоскость в  $\mathbb{R}^2$  (см. п.9.1). Множество, задаваемое системой неравенств 1-3, как видно из рис. 12.1, есть выпуклый<sup>3</sup> четырехугольник.

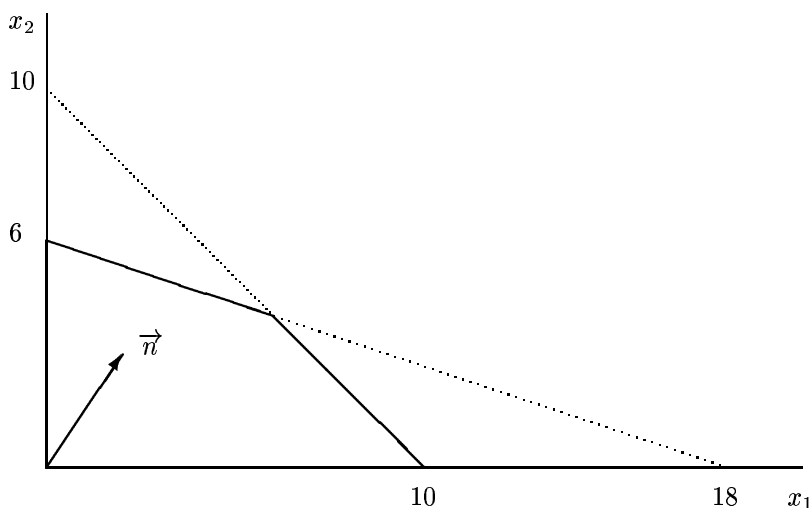


Рис. 12.1

Линии уровня функционала  $f$  – это семейство параллельных прямых (все они перпендикулярны отрезку  $\vec{n}$ , соответствующему вектору  $[0.5, 0.75]^T$ , (см. п.9.1). Из прямых этого семейства, пересекающих наш четырехугольник, следует выбрать ту, которая соответствует наибольшему значению  $f$ . Из рис.12.2 видно, что глобальный максимум  $f$  достигается в вершине четырехугольника с координатами  $x_1 = 6, x_2 = 4$ , а  $f_{\max} = 0.5 \cdot 6 + 0.75 \cdot 4 = 6$ .

<sup>1</sup>Леонид Витальевич КАНТОРОВИЧ (1912-1986) – советский математик и экономист, лауреат Нобелевской премии, член АН СССР и ряда зарубежных академий, один из основоположников математической экономики.

<sup>2</sup>Мы намеренно не уточняем, о каких единицах идет речь, чтобы сохранить коммерческую тайну.

<sup>3</sup>Множество называется выпуклым, если вместе с любыми двумя своими точками оно содержит соединяющий эти точки отрезок.

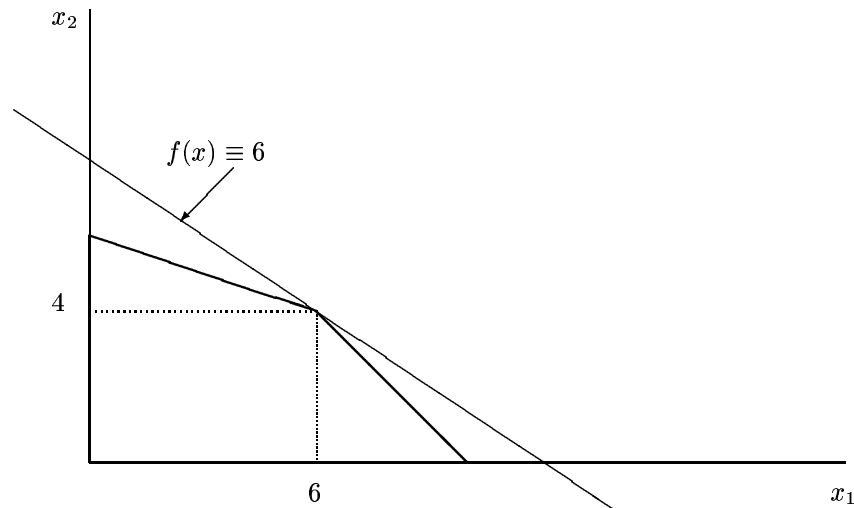


Рис. 12.2

### 12.2. Каноническая задача линейного программирования

Обобщим пример, рассмотренный в предыдущем пункте. Задача линейного программирования состоит в поиске глобального минимума (наименьшего значения) линейного функционала

$$f(x) = \langle x, c \rangle = c_1 x_1 + \dots + c_n x_n \quad (12.2.1)$$

на части  $\mathbb{R}^n$ , все точки которой удовлетворяют перечисленным ниже условиям:

$$a_{k1}x_1 + \dots + a_{kn}x_n \leq b_k \quad (k = 1, \dots, m_1); \quad (12.2.2)$$

$$a_{k1}x_1 + \dots + a_{kn}x_n = b_k \quad (k = m_1 + 1, \dots, m_2); \quad (12.2.3)$$

$$a_{k1}x_1 + \dots + a_{kn}x_n \geq b_k \quad (k = m_2 + 1, \dots, m). \quad (12.2.4)$$

Сделаем необходимые уточнения.

1. Существуют содержательные задачи (п.12.1), в которых функционал нужно не минимизировать, а максимизировать. Этот вариант, очевидно, укладывается в рассматриваемую схему с помощью замены вектора  $c$  на противоположный. Иначе говоря, *максимизация* функционала  $f(x) = \langle x, c \rangle$  – это то же самое, что *минимизация* функционала  $\varphi(x) = \langle x, -c \rangle$ .

2. Мы будем считать все переменные неотрицательными и выделим неравенства ( $x_k \geq 0$ ,  $k = 1, \dots, n$ ) в особую группу. Покажем, что это условие не является стесняющим. Действительно, если переменная  $x_k$  ограничена снизу ( $x_k \geq p$ ), то можно ввести новую переменную по формуле  $x_k^+ = x_k - p \geq 0$ . Если переменная  $x_k$  ограничена сверху ( $x_k \leq p$ ), то можно ввести новую переменную по формуле  $x_k^+ = p - x_k \geq 0$ . Если, наконец, переменная  $x_k$  не ограничена ни сверху, ни снизу, то (увеличивая количество переменных), положим  $x_k = x_k' - x_k''$ , где  $x_k' \geq 0$ ,  $x_k'' \geq 0$ .

Множество векторов из  $\mathbb{R}^n$  с *неотрицательными* координатами будем обозначать  $\mathbb{R}_+^n$ .

3. Мы будем считать все свободные члены в системе неравенств и уравнений (1.2.2) – (1.2.4) неотрицательными  $b \in \mathbb{R}_+^m$ . Этого всегда можно добиться, умножая при необходимости уравнение или неравенство на  $(-1)$ .

Можно показать, что<sup>4</sup> часть  $\mathbb{R}_+^n$ , задаваемая системой (12.2.2) – (12.2.4), есть

- 1) либо пустое множество (система несовместна),
- 2) либо точка,
- 3а) либо выпуклый *ограниченный* многогранник (пример – рис. 12.3),
- 3б) либо выпуклый *неограниченный* многогранник (пример – рис. 12.4).

<sup>4</sup>мы употребляем выражение "можно показать, что", если не можем или не хотим доказывать приводимое далее утверждение. Заинтересованный читатель может познакомиться с доказательством в более полном руководстве.

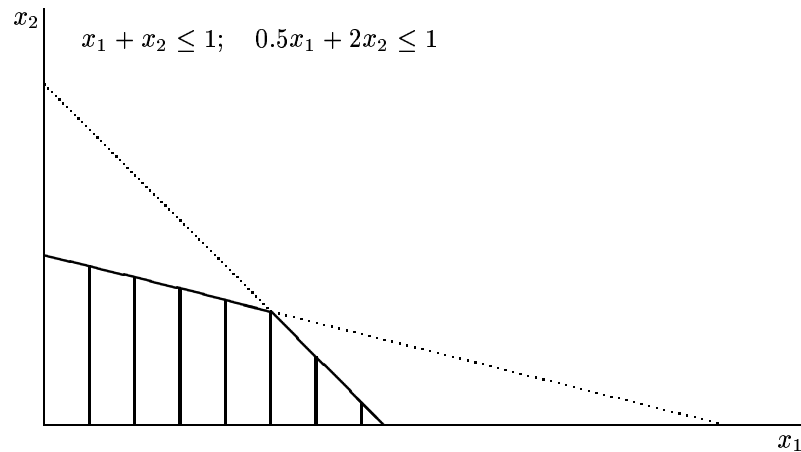


Рис. 12.3

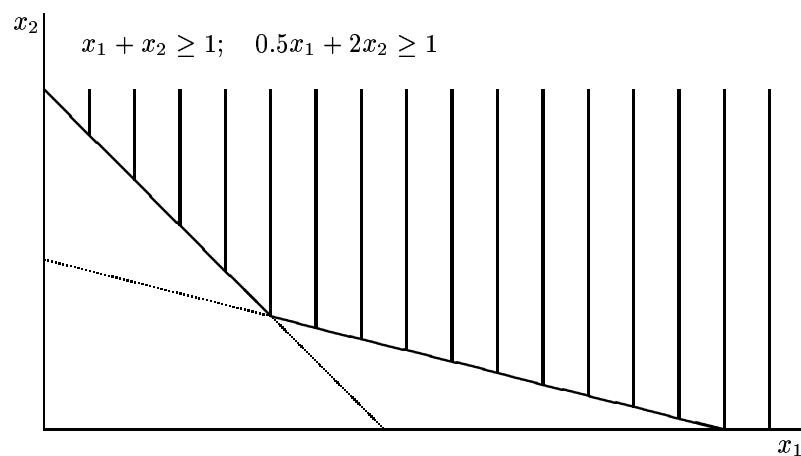


Рис. 12.4

В случае (1) задача линейного программирования не имеет решения;

В случае (2) решением задачи линейного программирования является, очевидно, единственное решение системы. Оба эти случая тривиальны.

Интерес представляют случаи (3а) и (3б).

В случае (3а) рассуждением, подобным проведенному в п. 12.1 *можно показать*, что глобальный минимум функционала существует и достигается хотя бы в одной из вершин многогранника.

В случае (3б) возможны два варианта: либо множество значений функционала не ограничено снизу, либо минимум существует и также достигается в одной из вершин многогранника.

Покажем теперь, что все ограничения-*неравенства*, кроме неравенств, гарантирующих неотрицательность координат вектора-решения, можно заменить ограничениями-*равенствами* (за счет увеличения количества переменных в задаче).

Действительно, можно ввести новую переменную  $x_{n+1} \geq 0$  и записать уравнение

$$a_{k1}x_1 + \dots + a_{kn}x_n + x_{n+1} = b_k,$$

из которого следует (12.2.2).

Аналогично, вводя новую переменную  $x_{n+1} \geq 0$  и записывая уравнение

$$a_{k1}x_1 + \dots + a_{kn}x_n - x_{n+1} = b_k,$$

получим (как следствие из него) (12.2.4).

Конечно, дополнительные переменные не должны входить в минимизируемый функционал (соответствующие координаты вектора  $s$  должны равняться нулю).

Сформулируем теперь так называемую *каноническую задачу линейного программирования*:

Минимизировать линейный функционал

$$f(x) = \langle x, c \rangle = c^T x$$

на части  $\mathbb{R}_+^n$ , все точки которой удовлетворяют системе линейных уравнений

$$Ax = b, \quad (12.2.5)$$

где  $A$  – матрица размера  $m \times n$ , а  $b \in \mathbb{R}_+^m$ .

**Замечания.** 1. Мы будем считать, что система (12.2.5) имеет бесконечно много решений в  $\mathbb{R}_+^n$ , иначе задача тривиальна. Можно также полагать, что ни одно из этих уравнений не является следствием других (иначе его можно просто вычеркнуть). Отсюда, кстати, следует, что  $m < n$ .

2. Мы по-прежнему обозначаем буквой  $n$  размерность пространства (количество переменных) и буквой  $m$  – количество ограничений-равенств. Следует однако помнить, что теперь количество переменных может быть *больше*, чем в исходной задаче – за счет дополнительных переменных, появляющихся при замене ограничений-неравенств ограничениями-равенствами. Количество ограничений может оказаться *меньше*, чем в исходной задаче – за счет вычленения уравнений, являющихся следствием оставшихся.

### 12.3. Преобразование канонической задачи линейного программирования

Из замечания 1 в конце п.12.2 следует, что при решении системы (12.2.5) значения некоторых  $m$  переменных однозначно определяются значениями оставшихся  $n - m$  переменных, которые могут быть заданы произвольно (здесь мы пока не учитываем, что все переменные должны быть неотрицательны). Отметим также, что указанные выше  $m$  переменных можно выбрать не единственным образом.

Будем считать, что определяемые переменные –  $x_1, \dots, x_m$ .

Разобьем матрицу  $A$ , вектор  $c$  и переменный  $x$  вектор на две части:

$$A = [B : N], \quad c = \begin{bmatrix} c^B \\ c^N \end{bmatrix}, \quad x = \begin{bmatrix} x^B \\ x^N \end{bmatrix},$$

где  $B$  – квадратная матрица порядка  $m$ ;  $N$  – матрица размера  $m \times (n - m)$ ;  $x^B, c^B$  – столбцы высоты  $m$ ;  $x^N, c^N$  – столбцы высоты  $n - m$ .

Теперь система (12.2.5) переписывается в виде

$$Bx^B + Nx^N = b, \quad (12.3.1)$$

а функционал (12.2.1) – в виде

$$f(x) = (c^B)^T \cdot x^B + (c^N)^T \cdot x^N. \quad (12.3.2)$$

Поскольку система (12.3.1) при каждом значении *однозначно* разрешима относительно  $x^B$ , то матрица  $B$  обратима, и мы получаем

$$x^B = B^{-1}b - B^{-1}Nx^N. \quad (12.3.3)$$

Подставляя полученный результат в (12.3.2), имеем

$$f(x) = (c^B)^T \cdot B^{-1}b + ((c^N)^T - (c^B)^T \cdot B^{-1} \cdot N) \cdot x^N.$$

Введем следующие обозначения:

$$\beta = B^{-1}b \quad (\text{столбец высоты } m),$$

$$\pi = (c^N)^T - (c^B)^T \cdot B^{-1} \cdot N \quad (\text{строка ширины } n - m).$$

Тогда получим так называемую *преобразованную задачу* линейного программирования:

Минимизировать функционал

$$\varphi(x^N) = (c^B)^T \cdot \beta + \pi \cdot x^N \quad (12.3.3)$$

при условиях

$$x^N \in \mathbb{R}_+^{n-m}, \quad x^B = \beta - B^{-1} \cdot N \cdot x^N \in \mathbb{R}_+^m. \quad (12.3.4)$$

Переменные – координаты вектора  $x^B$  – принято называть *базисными*, переменные – координаты вектора  $x^N$  – *небазисными*. Решение системы  $Ax = b$  называется *базисным*, если  $x^N = \theta_{n-m}$  ( $x^B = \beta$ ). Базисное

решение называется *допустимым*, если  $x^B = \beta \in \mathbb{R}_+^m$ . Если базисное решение допустимо и  $\pi^T \in \mathbb{R}_+^{n-m}$ , то это базисное решение доставляет функционалу искомый минимум, так как при  $x^N \in \mathbb{R}_+^{n-m}$

$$\varphi(x^N) = (c^B)^T \cdot \beta + \pi \cdot x^N \geq (c^B)^T \cdot \beta = \varphi(\theta_{n-m}).$$

Замечание. Столбец  $\beta$  и строка  $\pi$  зависят от выбора базисных переменных. Напомним, что этот выбор можно произвести не единственным образом.

Пример. Минимизировать функционал

$$f(x) = 3x_1 + x_2 + 2x_3$$

при условиях

$$x_1 + 2x_2 + 3x_3 = 6; \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Здесь  $n = 3$ ,  $m = 1$ ,  $A = [1, 2, 3]$ ,  $b = [6]$ ,  $c = [3, 1, 2]^T$ .

Если за базисную переменную принять  $x_1$ , то

$$B = [1], \quad N = [2, 3], \quad \beta = [6], \quad \pi = [1, 2] - 3 \cdot [2, 3] = [-5, -7].$$

Базисное решение  $[6, 0, 0]^T$  допустимо.

Если за базисную переменную принять  $x_3$ , то

$$B = [3], \quad N = [1, 2], \quad \beta = [2], \quad \pi = [3, 1] - \frac{2}{3} \cdot [1, 2] = \left[\frac{7}{3}, -\frac{1}{3}\right].$$

Базисное решение  $[0, 0, 2]^T$  также допустимо..

Если, наконец, за базисную переменную принять  $x_2$ , то

$$B = [2], \quad N = [1, 3], \quad \beta = [3], \quad \pi = [3, 2] - \frac{1}{2} \cdot [1, 3] = \left[\frac{5}{2}, \frac{1}{2}\right].$$

Базисное решение  $[0, 3, 0]^T$  допустимо и доставляет минимум функционалу, так как  $\pi^T \in \mathbb{R}_+^2$ .

Проиллюстрируем наш пример геометрически.

Система ограничений, состоящая из одного уравнения  $x_1 + 2x_2 + 3x_3 = 6$ , задает плоскость в  $\mathbb{R}^3$ . Пересечение этой плоскости с  $\mathbb{R}_+^3$  – треугольник (рис. 12.5). Легко видеть, что допустимые базисные решения  $[6, 0, 0]^T$ ,  $[0, 3, 0]^T$ ,  $[0, 0, 2]^T$  соответствуют вершинам этого треугольника. Как уже указывалось, минимальное значение функционала достигается хотя бы в одной из вершин. Сравнив значения функционала в вершинах  $f(6, 0, 0) = 18$ ,  $f(0, 3, 0) = 3$ ,  $f(0, 0, 2) = 4$ , видим, что базисное решение  $[0, 3, 0]^T$  действительно доставляет функционалу глобальный минимум.

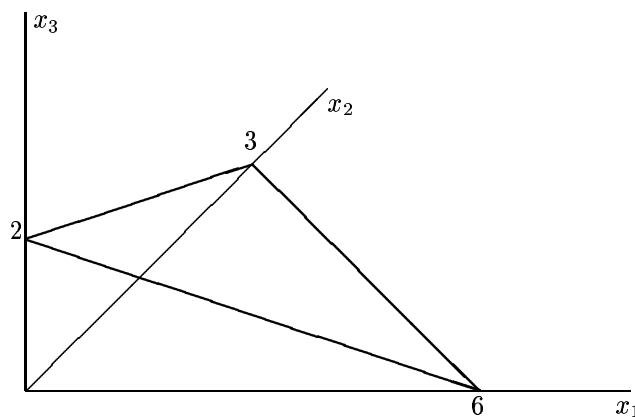


Рис. 12.5

#### 12.4. Понятие о симплекс-методе решения задачи линейного программирования

Этот метод, открытый в 1951 году Д. Данцигом<sup>5</sup>, представляет собой *конечный* алгоритм, который, начиная с некоторого *допустимого базисного* решения, строит новые *допустимые базисные решения*, обеспечивая *уменьшение* значения функционала.

Мы опишем один шаг симплекс-метода, не вдаваясь, естественно, в технологические подробности.

Итак, пусть имеется допустимое базисное решение

$$x = \begin{bmatrix} x^B \\ x^N \end{bmatrix},$$

где все координаты вектора  $x^B$  неотрицательны, а все координаты вектора  $x^N$  равны нулю.

Если  $\pi^T \in \mathbb{R}_+^{n-m}$  (все элементы строки  $\pi$  неотрицательны), то из (12.3.3)

$$\varphi(x^N) = (c^B)^T \cdot \beta + \pi \cdot x^N$$

видно, что наименьшее возможное значение функционала уже достигнуто

$$\min(\varphi(x^N)) = (c^B)^T \cdot \beta + \pi \theta_{n-m} = (c^B)^T \cdot \beta,$$

задача решена, и работа алгоритма заканчивается.

Если среди элементов строки  $\pi$  есть отрицательные, то, увеличивая соответствующие им координаты вектора  $x^N$ , мы будем двигаться в направлении убывания функционала. Скорость убывания пропорциональна значениям отрицательных элементов этой строки. Поэтому выберем из них наименьший (наибольший по модулю). Пусть его порядковый номер  $q$ . Будем увеличивать  $q$ -ю координату вектора  $x^N$  до тех пор, пока все координаты вектора  $x^B$  еще неотрицательны. Поскольку отлична от нуля только одна координата вектора  $x^N$ , (12.3.3) принимает вид

$$x^B = \beta - (B^{-1}N)^{(q)} x_q^N.$$

Если *все* элементы столбца  $(B^{-1}N)^{(q)}$  *неположительны*, то координаты вектора  $x^B$  остаются неотрицательными при любых положительных значениях  $x_q^N$ . Это значит, что множество значений функционала  $\varphi$  не ограничено снизу, и задача не имеет решения. Работа алгоритма заканчивается.

Если среди элементов столбца  $(B^{-1}N)^{(q)}$  есть *положительные*, то соответствующие им координаты вектора  $x^B$  будут убывать с увеличением  $x_q^N$ . Пусть  $s$  – номер той координаты вектора  $x^B$ , которая *первой обратится в нуль* в этом процессе. Тогда наибольшее возможное значение  $x_q^N$  равно

$$\max(x_q^N) = \frac{\beta_s}{((B^{-1}N)^{(q)})_s} = \min \left( \frac{\beta_j}{((B^{-1}N)^{(q)})_j} \right)$$

(минимум берется по тем индексам  $j$ , для которых  $((B^{-1}N)^{(q)})_j > 0$ !).

Теперь переменная  $x_q^N$  включается в состав базисных, а обнуленная переменная  $x_s^B$  исключается из него.

Дальнейшие действия можно было бы представить себе так: меняем местами столбцы  $B^{(s)}$  и  $N^{(q)}$  в матрице  $A$  и соответствующие координаты вектора  $s$ . Получим "новые" матрицы  $B$  и  $N$ . Вычислим новый вектор  $\beta$  и новую строку  $\pi$ . На полученном новом *допустимом* ( $x^B = \beta \in \mathbb{R}_+^m$ ) *базисном* ( $x^N = \theta_{n-m}$ ) решении функционал  $\varphi$  по *построению* имеет *меньшее* значение, чем на старом. Один шаг алгоритма закончен.

**Замечания.** 1. Поскольку количество допустимых базисных решений не превосходит количества различных наборов базисных переменных, а оно, в свою очередь, не превосходит количества сочетаний из  $n$  столбцов матрицы  $A$  по  $m$ , т.е.  $\frac{n!}{m!(n-m)!}$ , то за конечное число шагов алгоритм либо находит глобальный минимум функционала, либо выявляет его отсутствие.

2. Существуют эффективные вычислительные алгоритмы, реализованные в средах конечного пользователя и в библиотеках Фортрана.

<sup>5</sup> Джордж Бернارد ДАНЦИГ (род. 1914) – американский математик.

### 12.5. Построение начального допустимого базисного решения

Для начала работы алгоритма симплекс-метода необходимо иметь какое-нибудь допустимое базисное решение. Мы опишем один из возможных способов его построения.

Пусть требуется минимизировать функционал  $f(x) = \langle x, c \rangle$  при условиях

$$x \in \mathbb{R}_+^n, \quad Ax = b \in \mathbb{R}_+^m. \quad (12.5.1)$$

Рассмотрим вспомогательную задачу о минимизации линейного функционала  $\varphi(x, y) = y_1 + \dots + y_m$  при условиях

$$x \in \mathbb{R}_+^n, \quad y \in \mathbb{R}_+^m, \quad [A: I_m] \begin{bmatrix} x \\ y \end{bmatrix} = b. \quad (12.5.2)$$

Отметим, что при существовании решения задачи (12.5.1) должно существовать и решение задачи (12.5.2) с  $y = \theta$  (докажите это!).

Для задачи (12.5.2) одно допустимое базисное решение очевидно:  $y = b$ ,  $x = \theta_n$ . Поэтому можно применить к ней симплекс метод.

Поскольку функционал  $\varphi(x, y)$  ограничен снизу ( $y_1 + \dots + y_m \geq 0$ ), алгоритм за конечное число шагов даст решение вспомогательной задачи  $\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}$ .

Если окажется, что  $\tilde{y} = \theta_m$ , то  $\tilde{x}$  – допустимое базисное решение системы.

Если же  $\tilde{y} \neq \theta_m$ , то полученное противоречие свидетельствует о несовместности условий (12.5.1), и исходная задача не имеет решения.