Реализация симплекс-метода (математическое обоснование)

А. О. Махорин*

Март 2014 г.

1 Общие сведения

1.1 Рабочий формат ЛП-задачи

В рассматриваемой реализации используется так называемый *рабочий формат* задачи линейного программирования (ЛП-задачи):

$$z = c^T x + c_0 \to \min \tag{1.1}$$

$$Ax = b \tag{1.2}$$

$$l \le x \le u \tag{1.3}$$

где $x = (x_k)$ — вектор переменных, z — целевая функция, $c = (c_k)$ — вектор коэффициентов целевой функции, c_0 — постоянный член целевой функции, $A = (a_{ik})$ — матрица коэффициентов ограничений, $b = (b_i)$ — вектор правых частей ограничений, $l = (l_k)$ — вектор нижних границ переменных, $u = (u_k)$ — вектор верхних границ переменных.

Если нижняя (верхняя) граница переменной x_k отсутствует, то формально считается, что $l_k = -\infty$ ($u_k = +\infty$). Переменная, у которой отсутствуют обе границы, называется *свободной (неограниченной) переменной*. Если $l_k = u_k$, то переменная x_k считается *фиксированной*.

В дальнейшем предполагается, что число переменных равно n, а число ограничений-равенств (1.2) равно m, поэтому матрица A имеет m строк и n столбцов. Также предполагается, что эта матрица имеет полный строчный ранг $\mathrm{rank}(A) = m$, т. е. все строки A линейно независимы, откуда, в частности, следует, что $m \le n$.

*Кафедра прикладной информатики, Московский авиационный институт, Москва, Россия. E-mail: <mao@gnu.org>.

1.2 Базисные решения

При решении ЛП-задачи симплекс-методом среди всевозможных допустимых решений, т. е. точек, удовлетворяющих всем ограничениям, особую роль играют так называемые *базисные решения*. Геометрически допустимые базисные решения соответствуют вершинам (крайним точкам) выпуклого многогранного множества[1] допустимых точек ЛП-задачи, где каждая вершина определяется как точка пересечения граней (гиперплоскостей), соответствующих активным[2] линейно независимым ограничениям.

Так как для ЛП-задачи (1.1)—(1.3) предполагается, что все ограничения (1.2) линейно независимы (матрица A имеет полный строчный ранг), то, являясь равенствами, все эти ограничения будут активны в любом базисном решении. Следовательно, для однозначного определения базисного решения как точки в n-мерном пространстве переменных $x = (x_k)$ к m ограничениям-равенствам (1.2) необходимо добавить еще $n - m$ активных ограничений из числа границ переменных (1.3).

Переменные, границы которых являются активными ограничениями в базисном решении, называются *небазисным (независимыми) переменными*. Остальные переменные с неактивными границами называются *базисными (зависимыми) переменными*. Поскольку нижняя и верхняя границы небазисной переменной не могут быть активны одновременно,[3] то в любом базисном решении всегда будет $n - m$ небазисных и m базисных переменных.

Таким образом, базисное решение ЛП-задачи (1.1)—(1.3) однозначно определено, если указано, какие переменные являются базисными, а какие небазисными, и для каждой небазисной переменной, имеющей обе (нижнюю и верхнюю) границы, также указано какая именно из этих границ активна. При этом требуется, чтобы все n активных ограничений были линейно независимыми.

Перейдем к алгебраическому описанию базисных решений. Пусть

$$\Pi x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \ \Pi c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}, \ \Pi l = \begin{pmatrix} l_B \\ l_N \end{pmatrix}, \ \Pi u = \begin{pmatrix} u_B \\ u_N \end{pmatrix}, \qquad (1.4)$$

где Π — подходящая перестановочная матрица, $x_B = [(x_B)_i]$ — вектор базисных переменных, $x_N = [(x_N)_j]$ — вектор небазисных переменных, $c_B = [(c_B)_i]$ и $c_N = [(c_N)_j]$ — векторы коэффициентов целевой функции, $l_B = [(l_B)_i]$ и $l_N = [(l_N)_j]$ — векторы нижних границ, $u_B = [(u_B)_i]$

[1] Если это множество ограничено, то оно представляет собой выпуклый многогранник.

[2] В данном случае более правильным было бы называть эти ограничения *связывающими*.

[3] Случай фиксированной переменной, т. е. когда $l_k = u_k$, можно рассматривать как предельный случай двустороннего ограничения $l_k - \varepsilon \le x_k \le u_k + \varepsilon$ при $\varepsilon \to 0$.

и $u_N = [(u_N)_j]$ — векторы верхних границ, соответственно, базисных и небазисных переменных. Соотношение (1.4) позволяет представить ограничения-равенства (1.2) в следующем эквивалентном виде:

$$A\Pi^T \Pi x = b \quad \Leftrightarrow \quad (B \,|\, N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \quad \Leftrightarrow \quad Bx_B + Nx_N = b, \tag{1.5}$$

где

$$(B \,|\, N) = A\Pi^T. \tag{1.6}$$

Здесь B — квадратная матрица порядка m, называемая *базисной матрицей* (или просто *базисом*). Эта матрица составлена из столбцов исходной матрицы коэффициентов ограничений A, соответствующих базисным переменным. Матрица N имеет m строк и $n-m$ столбцов, которыми являются столбцы A, соответствующие небазисным переменным.

Пусть $f = (f_j)$ — вектор активных границ небазисных переменных, т. е. $f_j = (l_N)_j$ или $f_j = (u_N)_j$. Тогда базисное решение определяется следующей системой n уравнений, составленной из m активных ограничений-равенств (1.2) в форме (1.5) и $n-m$ активных границ небазисных переменных:

$$\begin{cases} Bx_B + Nx_N = b \\ x_N = f \end{cases} \tag{1.7}$$

Так как все активные ограничения предполагаются линейно независимыми, то матрица системы (1.7) невырожденная, откуда следует, что базисная матрица B также невырожденная. Это позволяет разрешить систему ограничений-равенств (1.2) в форме (1.5) относительно базисных переменных, чтобы явно выразить их через небазисные переменные:

$$x_B = -B^{-1}Nx_N + B^{-1}b \quad \Leftrightarrow \quad x_B = \Xi x_N + g, \tag{1.8}$$

где

$$\Xi = -B^{-1}N, \tag{1.9}$$

$$g = B^{-1}b. \tag{1.10}$$

Здесь $\Xi = (\xi_{ij})$ — $m \times (n-m)$-матрица, называемая *симплекс-таблицей*, $g = (g_i)$ — m-вектор, называемый *преобразованным вектором правых частей ограничений.*

Определения (1.9) и (1.10) с учетом соотношения (1.4) позволяют записать базисное решение как решение системы (1.7):

$$x = \Pi^T \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \Pi^T \begin{pmatrix} \beta \\ f \end{pmatrix}, \tag{1.11}$$

где

$$\beta = (\beta_i) = -B^{-1}Nf + B^{-1}b = \Xi f + g \tag{1.12}$$

есть m-вектор значений базисных переменных в базисном решении.

Примечание. Может случиться, что небазисной является свободная переменная[4] $(x_N)_j = x_k$, у которой отсутствуют обе нижняя и верхняя границы ($l_k = -\infty$ и $u_k = +\infty$). Тогда мы можем формально рассматривать такую переменную как сумму двух переменных $x_k = x'_k + x''_k$, где $x'_k \geq 0$ (неотрицательная переменная), $x''_k \leq 0$ (неположительная переменная). Переменные x'_k и x''_k не могут быть базисными одновременно, так как в противном случае в базисной матрице B были бы два одинаковых, а значит, линейно зависимых столбца. Поэтому, если x_k является базисной и ее текущее значение неотрицательно (неположительно), то это эквивалентно тому, что базисной является переменная x'_k (соответственно, x''_k), а переменная x''_k (соответственно, x'_k) является небазисной и поэтому находится на своей активной нулевой границе. Следовательно, если x_k является небазисной, то это эквивалентно тому, что обе переменные x'_k и x''_k небазисные, а значит, формально можно считать, что в этом случае значение переменной $(x_N)_j = x_k$ в базисном решении равно нулю, как если бы эта переменная имела активную нулевую границу.

Базисное решение является допустимым, если значения всех переменных удовлетворяют всем ограничениям ЛП-задачи. Так как соотношение (1.8) эквивалентно исходной системе ограничений-равенств (1.2), то любое базисное решение по определению удовлетворяет этим ограничениям. Кроме того, значения небазисных переменных в любом базисном решении совпадают с их границами, поэтому нарушение границ (1.3) возможно только для базисных переменных. Таким образом, базисное решение (1.11) является допустимым только если $l_B \leq \beta \leq u_B$, т. е. если значения всех базисных переменных находятся внутри соответствующих границ.

Всякое недопустимое базисное решение характеризуется тем, что хотя бы одна базисная переменная в этом решении нарушает свою нижнюю или верхнюю границу. Геометрически недопустимые базисные решения также представляют собой точки пересечения граней (гиперплоскостей), соответствующих активным линейно независимым ограничениям, однако эти точки лежат вне выпуклого многогранного множества допустимых решений ЛП-задачи.

[4] Такую переменную более правильно было бы называть *супербазисной* по аналогии с супербазисными переменными в задачах с нелинейной целевой функцией.

1.3 Симплекс-метод

Симплекс-метод — это численный метод решения ЛП-задач, разработанный Дж. Данцигом (G. B. Dantzig) в 1947 г.[5]

С геометрической точки зрения общая идея симплекс-метода достаточно проста и заключается в следующем. Предположим, что мы находимся в некоторой вершине многогранного множества допустимых решений ЛП-задачи. Из этой вершины исходят *ребра*, ведущие в *смежные* вершины. Если движение вдоль какого-либо ребра приводит к улучшению целевой функции, то мы переходим в соответствующую смежную вершину с лучшим значением целевой функции, после чего повторяем все сначала до тех пор, пока это возможно. Поскольку число вершин многогранного множества с конечным числом граней конечно, то после выполнения конечного[6] числа шагов мы придем в вершину, из которой переход в смежную вершину с лучшим значением целевой функции невозможен. Вершина, обладающая указанным свойством, будет оптимальным решением ЛП-задачи.

Рассмотрим алгебраическое описание симплекс-метода применительно к ЛП-задаче в формате (1.1)—(1.3).

Пусть у нас имеется допустимое базисное решение (1.11), соответствующее текущей вершине многогранного множества допустимых решений. Движение вдоль ребра по направлению к смежной вершине характеризуется тем, что какое-либо одно активное ограничение перестает быть активным, в то время как остальные активные ограничения остаются активными. Так как ограничения-равенства (1.2) активны в любом базисным решении, то указанным ограничением может быть только ограничение вида (1.3), т. е. активная граница какой-либо небазисной переменной $(x_N)_j$. Таким образом, движение вдоль ребра означает, что соответствующая этому ребру небазисная переменная начинает изменяться в допустимом направлении: увеличиваться, если активной была ее нижняя граница, или уменьшаться, если активной была ее верхняя граница.

Предположим, что небазисная переменная $(x_N)_j$ изменяется в допустимом направлении. В соответствии с (1.8) это приведет к изменению базисных переменных, а также целевой функции. Чтобы оценить суммарное изменение целевой функции (1.1), обусловленое изменением небазисной и базисных переменных, запишем ее в следующем виде, используя соотношение (1.4):

$$z = c^T \Pi^T \Pi x + c_0 = \begin{pmatrix} c_B \\ c_N \end{pmatrix}^T \begin{pmatrix} x_B \\ x_N \end{pmatrix} + c_0 = c_B^T x_B + c_N^T x_N + c_0, \quad (1.13)$$

[5] Впервые симплекс-метод был опубликован в работе [1]. См. также [2].

[6] При условии, что приняты необходимые меры, предотвращающие зацикливание, которое возможно, если имеются кратные (совпадающие) вершины.

после чего подставим в (1.13) выражение для x_B из (1.8). В результате получим:

$$\begin{aligned} z &= c_B^T(-B^{-1}Nx_N + B^{-1}b) + c_N^T x_N + c_0 = \\ &= (c_N^T - c_B^T B^{-1}N)x_N + c_B^T B^{-1}b + c_0 = \\ &= (c_N - N^T B^{-T} c_B)^T x_N + (B^{-T}c_B)^T b + c_0 = \\ &= d^T x_N + d_0, \end{aligned} \tag{1.14}$$

где

$$d = c_N + \Xi^T c_B = c_N - N^T \pi, \tag{1.15}$$

$$\pi = B^{-T} c_B, \tag{1.16}$$

$$d_0 = \pi^T b + c_0. \tag{1.17}$$

Здесь $d = (d_j)$ — вектор так называемых *относительных оценок небазисных переменных*,[7] где относительная оценка d_j показывает суммарное (т. е. прямое и косвенное через базисные переменные) влияние небазисной переменной $(x_N)_j$ на целевую функцию при условии, что остальные небазисные переменные не изменяются. Вектор $\pi = (\pi_i)$ — это вектор так называемых *оценок правых частей ограничений* (или *симплексных множителей*),[8] где оценка правой части π_i показывает влияние изменения правой части i-го ограничения b_i на целевую функцию. Скалярная величина d_0 есть постоянный член преобразованной целевой функции (1.14).

В случае минимизации движение вдоль выбранного ребра должно приводить к уменьшению целевой функции. Так как выбор ребра соответствует выбору небазисной переменной, это означает, что мы можем выбрать любую небазисную переменную с активной нижней границей, относительная оценка которой отрицательна, или любую небазисную переменную с активной верхней границей, относительная оценка которой положительна, поскольку изменение такой переменной в допустимом направлении даст уменьшение целевой функции. В случае свободной небазисной переменной достаточно, чтобы ее относительная оценка была ненулевой, так как свободная переменная может изменяться в любом направлении. Если при этом не найдется ни одной подходящей небазисной переменной, то текущее базисное решение соответствует минимуму целевой функции (является оптимальным). Заметим, что обычно имеется несколько подходящих небазисных переменных, поэтому возникает проблема выбора наиболее подходящей небазисной переменной.

[7] Компоненты вектора d суть значения множителей Лагранжа для активных границ небазисных переменных.

[8] Компоненты вектора π суть значения множителей Лагранжа для ограничений-равенств (1.2).

Допустим, что выбрана подходящая небазисная переменная $(x_N)_q$ и эта переменная начинает изменяться в допустимом направлении (возрастать или убывать), при этом остальные небазисные переменные остаются на своих активных границах. Тогда в соответствии с (1.8) начнут изменяться базисные переменные x_B. Так как текущее базисное решение предполагается допустимым, то в начальный момент, когда $(x_N)_q$ находится на своей активной границе, все базисные переменные будут находиться внутри своих границ. Понятно, что если в процессе изменения $(x_N)_q$ некоторая базисная переменная $(x_B)_p$ первой достигнет своей (нижней или верхней) границы, либо сама небазисная переменная $(x_N)_q$ первой достигнет своей противоположной границы, то это будет означать, что мы достигли смежной вершины и дальнейшее изменение $(x_N)_q$ в том же направлении невозможно, так как это приведет к нарушению границы соответствующей переменной, а значит, к выходу за пределы допустимого множества.

Чтобы найти переменную, определяющую смежное базисное решение, положим

$$(x_N)_q = f_q + s\theta, \tag{1.18}$$

где f_q — активная граница небазисной переменной $(x_N)q$ в текущем базисном решении, $s = -\text{sign } d_q$ ($s = +1$, если $d_q < 0$, т. е. $(x_N)_q$ возрастает, или $s = -1$, если $d_q > 0$, т. е. $(x_N)_q$ убывает), $\theta \geq 0$ — скалярный параметр луча, который исходит из текущей вершины и направление которого совпадает с направлением ребра, соответствующего переменной $(x_N)_q$. Заметим, что увеличение параметра θ начиная с нуля соответствует изменению $(x_N)_q$ в допустимом направлении начиная с ее активной границы f_q. Поскольку остальные небазисные переменные не изменяются, то с учетом (1.8) и (1.12) отдельная базисная переменная $(x_B)_i$ будет зависеть от θ следующим образом:

$$(x_B)_i = \beta_i + \xi_{iq} s\theta, \tag{1.19}$$

где β_i — значение $(x_B)_i$ в текущем базисном решении, ξ_{iq} — элемент симплекс-таблицы (1.9). Очевидно, что если $\xi_{iq}s < 0$, то $(x_B)_i$ убывает вдоль луча, а если $\xi_{iq}s > 0$, то $(x_B)_i$ возрастает вдоль луча. Поэтому $(x_B)_i$ достигнет своей нижней границы $(l_B)_i$ или верхней границы $(u_B)_i$ при $\theta = \theta_i \geq 0$, где

$$\theta_i = \begin{cases} \dfrac{(l_B)_i - \beta_i}{\xi_{iq}s}, & \text{если } \xi_{iq}s < 0 \\ \dfrac{(u_B)_i - \beta_i}{\xi_{iq}s}, & \text{если } \xi_{iq}s > 0 \end{cases} \tag{1.20}$$

а значит, *первой* достигнет своей границы та базисная переменная $(x_B)_p$, для которой

$$\theta_p = \min_i \theta_i. \tag{1.21}$$

Кроме того, небазисная переменная $(x_N)_q$ достигнет своей противоположной границы при $\theta = \theta_0 \geq 0$, где

$$\theta_0 = |(l_N)_q - (u_N)_q|. \tag{1.22}$$

Поэтому луч останется в допустимом множестве при $0 \leq \theta \leq \theta_{\min}$, где

$$\theta_{\min} = \min(\theta_p, \theta_0). \tag{1.23}$$

Таким образом, величина $\theta_{\min}$ позволяет идентифицировать переменную, которая первой достигает своей границы и тем самым определяет смежное базисное решение: если $\theta_{\min} = \theta_p$, то это базисная переменная $(x_B)_p$, а если $\theta_{\min} = \theta_0$, то это небазисная переменная $(x_N)_q$. Нужно также заметить, что возможен случай $\theta_{\min} = \infty$, который означает, что ЛП-задача имеет неограниченный оптимум (минимум), т. е. в этом случае возможно неограниченное уменьшение целевой функции вдоль соответствующего луча.

Если $\theta_{\min} = \theta_p$, то в смежном базисном решении та (нижняя или верхняя) граница переменной $(x_B)_p$, которая препятствует дальнейшему изменению $(x_N)_q$, т. е. которая определяет величину θ_p, становится активной, а текущая активная граница f_q переменной $(x_N)_q$, наоборот, перестает быть активной. Другими словами, в смежном базисе переменные $(x_B)_p$ и $(x_N)_q$ меняются ролями: $(x_B)_p$ становится небазисной (выходит из базиса), а $(x_N)_q$ становится базисной (входит в базис). Если же $\theta_{\min} = \theta_0$, то в смежном базисе переменная $(x_N)_q$ остается небазисной, однако активной становится ее противоположная граница. Зная величину $\theta_{\min}$ можно также определить величину приращения целевой функции при переходе от текущего базиса к смежному:

$$\Delta z = -|d_q \theta_{\min}| \leq 0. \tag{1.24}$$

Выбор базисной переменной может оказаться неоднозначным, если несколько переменных одновременно (т. е. при одном и том же значении $\theta = \theta_{\min}$) достигают своих границ, причем эта неоднозначность обычно усугубляется ситуацией $\theta_{\min} = 0$. В последнем случае при переходе к смежному базису целевая функция не изменяется, поэтому, если не принимать специальных мер, возможно зацикливание метода, когда происходит возврат к одному из предыдущих базисных решений.

В заключение рассмотрим принципиальную схему симплекс-метода применительно к ЛП-задаче (1.1)—(1.3), полагая, что задано некоторое начальное допустимое базисное решение (указано, какие переменные являются базисными, какие небазисными, и для небазисных переменных указаны их активные границы).

СИМПЛЕКС-МЕТОД

Шаг 1. Вычислить $\beta = (\beta_i)$ — вектор текущих значений базисных переменных $x_B = [(x_B)_i]$ (1.12), $d = (d_j)$ — вектор относительных оценок небазисных переменных $x_N = [(x_N)_j]$ (1.16).

Шаг 2. Выбрать подходящую небазисную переменную $(x_N)_q$, изменение которой в допустимом направлении уменьшает целевую функию, т. е. относительная оценка d_q которой имеет «неправильный» знак. Небазисная переменная $(x_N)_j$ считается подходящей, если:

$d_j < 0$ и $(x_N)_j$ имеет активную нижнюю границу;

$d_j > 0$ и $(x_N)_j$ имеет активную верхнюю границу;

$d_j \neq 0$ и $(x_N)_j$ является свободной (неограниченной) переменной.

Фиксированные небазисные переменные не могут изменяться и поэтому не включаются в число подходящих.

Если выбор невозможен, то СТОП — текущее базисное решение является оптимальным.

Шаг 3. Определить $\theta_{\min}$ — абсолютную величину приращения выбранной небазисной переменной $(x_N)_q$ в смежном базисе (1.23).

Если $\theta_{\min} = \infty$, то СТОП — ЛП-задача имеет неограниченный минимум.

Шаг 4. Если $\theta_{\min} = \theta_p$ (1.21), то пометить переменную $(x_B)_p$ как небазисную с активной нижней или верхней границей в зависимости от выбора $\theta_i = \theta_p$ (1.20), а переменную $(x_N)_q$ пометить как базисную. Если же $\theta_{\min} = \theta_0$ (1.22), то изменить текущую активную границу переменной $(x_N)_q$ на противоположную.

Вернуться к шагу 1. ■

1.4 Табличный симплекс-метод

Табличный симплекс-метод — это вариант симплекс-метода, в котором для определения всех необходимых величин используется таблица (двумерный массив), соответствующая так называемой *преобразованной* (или *текущей*) *задаче*.

Преобразованная задача получается из исходной ЛП-задачи посредством эквивалентного линейного преобразования,[9] которое применяется для того, чтобы явно выразить целевую функцию и базисные переменные через небазисные переменные.

Так, для ЛП-задачи в формате (1.1)—(1.3) преобразованная задача имеет следующий формат:

$$z = d^T x_N + d_0 \to \min \tag{1.25}$$

$$x_B = \Xi x_N + g \tag{1.26}$$

$$l_B \le x_B \le u_B \tag{1.27}$$

$$l_N \le x_N \le u_N \tag{1.28}$$

где x_B — вектор базисных переменных, x_N — вектор небазисных переменных, d — вектор относительных оценок небазисных переменных (1.15), d_0 — постоянный член преобразованной целевой функции (1.17), Ξ — симплекс-таблица (1.9), g — преобразованный вектор правых частей ограничений (1.10).

Существенная особенность преобразованной задачи состоит в том, что ее основные компоненты (d, d_0, Ξ, g) относятся к текущему базису, поэтому всякий раз при переходе к смежному базису необходимо выполнять пересчет этих компонент.

Главным преимуществом табличного симплекс-метода является простота его реализации, поскольку вся необходимая информация доступна в явном виде.[10] Однако для практического решения ЛП-задач табличный симплекс-метод в настоящее время почти не применяется в основном из-за его низкой вычислительной эффективности, особенно при решении разреженных ЛП-задач высокой размерности. Например, на каждой итерации табличного симплекс-метода необходимо выполнять

[9] Это преобразование, которое можно выполнить, используя метод исключения Гаусса—Жордана, представляет собой умножение B^{-1} слева на исходную матрицу коэффициентов ограничений (1.6) $A\Pi^T = (B \,|\, N)$, в результате чего получается матрица $B^{-1}(B \,|\, N) = (I \,|\, B^{-1}N) = (I \,|\, -\Xi)$, соответствующая (1.26).

[10] Наиболее ярко это проявляется для ЛП-задач в стандартном формате, где вместо границ общего вида (1.3) используются условия неотрицательности $x \ge 0$: в этом случае все небазисные переменные имеют нулевые активные границы, поэтому текущее значение целевой функции равно d_0, а текущие значения базисных переменных равны элементам вектора g.

пересчет всей симплекс-таблицы Ξ, хотя для выбора базисной переменной нужен всего один столбец этой матрицы. При этом даже если исходная матрица A является достаточно разреженной, заполнение симплекс-таблицы $\Xi = -B^{-1}N$ в общем случае может быть гораздо выше.

В настоящее время наиболее широко используется вариант, называемый *модифицированным симплекс-методом* (см. следующий раздел). В этом варианте компоненты преобразованной задачи не хранятся в явном виде, а все используемые величины вычисляются только по мере необходимости, что позволяет существенно повысить вычислительную эффективность метода. Тем не менее, модифицированный симплекс-метод также использует компоненты преобразованной задачи, поэтому имеет смысл рассмотреть здесь формулы табличного симплекс-метода для пересчета компонент преобразованной задачи (1.25)—(1.28) при переходе к смежному базису, поскольку эти формулы можно использовать для вывода формул пересчета других величин, связанных с текущим базисным решением.

Чтобы упростить вывод указанных формул, заметим, что строку целевой функции (1.25) можно рассматривать как строку свободной базисной переменной $(x_B)_0 = z$. Аналогично, преобразованный вектор правых частей ограничений g можно рассматривать как столбец фиксированной небазисной переменной $(x_N)_0 = 1$. Это позволяет объединить основные компоненты (d, d_0, Ξ, g) в *расширенную симплекс-таблицу*:[11]

$$\Xi^* = (\xi_{ij}) = \begin{pmatrix} d_0 & d^T \\ g & \Xi \end{pmatrix}, \tag{1.29}$$

а также объединить равенства (1.25) и (1.26) в одно равенство:

$$x_B = \Xi^* x_N. \tag{1.30}$$

Допустим, что базисная переменная $(x_B)_p$ становится небазисной и занимает место небазисной переменной $(x_N)_q$, которая, в свою очередь, становится базисной и занимает место переменной $(x_B)_p$. Нас интересуют формулы пересчета элементов расширенной симплекс-таблицы Ξ^* при переходе к смежному базису.

Чтобы выразить новую базисную переменную $\overline{(x_B)}_p = (x_N)_q$ через новый набор небазисных переменных, включающий новую небазисную переменную $\overline{(x_N)}_q = (x_B)_p$, преобразуем сначала p-ю (ведущую) строку системы равенств (1.30):

$$(x_B)_p = \sum_{j=0}^{n-m} \xi_{pj}(x_N)_j = \xi_{pq}(x_N)_q + \sum_{j \neq q} \xi_{pj}(x_N)_j,$$

[11] Эта расширенная симплекс-таблица, представленная в виде двумерного массива, является той основной структурой данных, которая используется в табличном симплекс-методе.

откуда

$$(x_N)_q = \frac{1}{\xi_{pq}}(x_B)_p - \sum_{j\neq q}\frac{\xi_{pj}}{\xi_{pq}}(x_N)_j. \tag{1.31}$$

Исключим теперь $(x_N)_q$ из остальных строк системы равенств (1.30), для чего подставим $(x_N)_q$ из (1.31) в каждую i-ю строку ($i \neq p$):

$$\begin{aligned}(x_B)_i &= \sum_{j=0}^{n-m}\xi_{ij}(x_N)_j = \xi_{iq}(x_N)_q + \sum_{j\neq q}\xi_{ij}(x_N)_j = \\ &= \frac{\xi_{iq}}{\xi_{pq}}(x_B)_p - \sum_{j\neq q}\frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}(x_N)_j + \sum_{j\neq q}\xi_{ij}(x_N)_j = \\ &= \frac{\xi_{iq}}{\xi_{pq}}(x_B)_p + \sum_{j\neq q}\left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)(x_N)_j.\end{aligned} \tag{1.32}$$

Коэффициенты при переменных в равенствах (1.31) и (1.32) являются элементами новой расширенной симплекс-таблицы $\overline{\Xi}^* = (\overline{\xi}_{ij})$ для смежного базиса, выраженные через элементы исходной расширенной симплекс-таблицы $\Xi^* = \xi_{ij}$ для текущего базиса. Таким образом, получаем следующие формулы пересчета:

$$\overline{\xi}_{pq} = \frac{1}{\xi_{pq}}, \tag{1.33}$$

$$\overline{\xi}_{pj} = -\frac{\xi_{pj}}{\xi_{pq}} = -\overline{\xi}_{pq}\xi_{pj}, \quad j \neq q. \tag{1.34}$$

$$\overline{\xi}_{iq} = \frac{\xi_{iq}}{\xi_{pq}} = \overline{\xi}_{pq}\xi_{iq}, \quad i \neq p. \tag{1.35}$$

$$\overline{\xi}_{ij} = \xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}} = \xi_{ij} - \overline{\xi}_{iq}\xi_{pj} = \xi_{ij} + \xi_{iq}\overline{\xi}_{pj}, \quad i \neq p, j \neq q. \tag{1.36}$$

Нулевая строка расширенной симплекс-таблицы (1.29) соответствует строке целевой функции в преобразованной задаче (1.25), т. е. $z = (x_B)_0$. Поэтому формулы пересчета относительных оценок $d_j = \xi_{0j}$ и постоянного члена $d_0 = \xi_{00}$ совпадают с формулами (1.35) и (1.36) при $i = 0$:

$$\overline{d}_q = \frac{d_q}{\xi_{pq}} = \overline{\xi}_{pq}d_q, \tag{1.37}$$

$$\overline{d}_j = d_j - \frac{d_q\xi_{pj}}{\xi_{pq}} = d_j - \overline{d}_q\xi_{pj} = d_j + d_q\overline{\xi}_{pj}, \quad j \neq q. \tag{1.38}$$

2 Модифицированный симплекс-метод

Модифицированный симплекс-метод [3, 4, 5] — это вариант симплекс-метода, в котором для вычисления всех необходимых величин используется подходящее представление текущей базисной матрицы B (1.6), позволяющее эффективно выполнять основные операции с этой матрицей (подразд. 2.1). В отличие от табличного симплекс-метода, (см. подразд. 1.4), преобразованная задача (1.25)—(1.28), за исключением ее отдельных компонент, в явном виде не хранится, что позволяет существенно повысить вычислительную эффективность симплекс-метода, особенно при решении разреженных ЛП-задач большой размерности.

2.1 Факторизация базисной матрицы

Представление текущей базисной матрицы B, используемое в модифицированном симплекс-методе, обычно (хотя и не обязательно) имеет вид матричной факторизации, где B представлена в виде произведения матриц специальной структуры.[12] Поэтому указанное представление называется в дальнейшем *факторизацией базисной матрицы.*

С точки зрения симплекс-метода вид конкретной факторизации базисной матрицы является несущественной. Однако предполагается, что используемая факторизация обеспечивает эффективное выполнение следующих четырех основных операций:

1. *Начальное вычисление факторизации.* Исходными данными этой операции является заданная матрица B. Результатом является вычисленная факторизация этой матрицы.

2. *Прямое преобразование (FTRAN).* Эта операция представляет собой решение системы $Bx = y$ (или, что то же самое, вычисление вектора $x = B^{-1}y$) для заданного вектора y.[13]

3. *Обратное преобразование (BTRAN).* Эта операция представляет собой решение системы $B^T x = y$ (или, что то же самое, вычисление вектора $x = B^{-T}y$) для заданного вектора y.

4. *Пересчет факторизации после замены столбца.* Исходными данными этой операции является факторизация текущей матрицы B и заданный вектор, который заменяет некоторый столбец в текущей матрице, что дает новую матрицу $\overline{B}$. Результатом операции является факторизация новой матрицы $\overline{B}$.

Примечание. Конкретная реализация факторизации базисной матрицы обычно ограничивает число пересчетов факторизации (например, не более 50 или 100), что обусловлено, в основном, техническими причинами (накопление вычислительных погрешностей, переполнение массивов, чрезмерное заполнение, и т. п.). Однако это ограничение не является

[12] Примером факторизации может служить треугольное разложение.

[13] Здесь x и y — это произвольные векторы, не связанные с ЛП-задачей.

критическим для модифицированного симплекс-метода, так как всегда имеется возможность выполнить начальное вычисление факторизации.

Реализация факторизации базисной матрицы практически не связана с реализацией симплекс-метода и представляет собой самостоятельную проблему, которая здесь не обсуждается.

В заключение рассмотрим формулы преобразования базисной матрицы при замене столбца. Эти формулы не зависят от вида конкретной факторизации и поэтому их можно использовать для пересчета других величин, связанных с базисной матрицей.

Итак, пусть p-й столбец матрицы B заменяется столбцом v, в результате чего получается матрица $\overline{B}$.

Так как B — невырожденная матрица, то имеет место равенство

$$B^{-1}B = I. \tag{2.1}$$

Определим матрицу H следующим образом:

$$B^{-1}\overline{B} = H. \tag{2.2}$$

Поскольку $\overline{B}$ отличается от B только p-м столбцом, то из сопоставления (2.1) и (2.2) следует, что H отличается от единичной матрицы I также только p-м столбцом и поэтому имеет вид:

$$H = \begin{pmatrix} 1 & & h_{1p} & & \\ & \ddots & \vdots & & \\ & 1 & h_{p-1,p} & & \\ & & h_{pp} & & \\ & & h_{p+1,p} & 1 & \\ & & \vdots & & \ddots \\ & & h_{mp} & & 1 \end{pmatrix}, \tag{2.3}$$

где p-й столбец этой матрицы $H_p = (h_{ip})$ удовлетворяет равенству

$$B^{-1}v = H_p. \tag{2.4}$$

Умножая слева B на обе части равенства (2.2), получим основную формулу преобразования базисной матрицы:

$$\overline{B} = BH. \tag{2.5}$$

Формулы преобразования для обратных матриц непосредственно следуют из (2.5):

$$\overline{B}^{-1} = H^{-1}B^{-1}, \tag{2.6}$$

$$\overline{B}^{-T} = B^{-T}H^{-T}, \tag{2.7}$$

где

$$H^{-1} = \begin{pmatrix} 1 & & & -h_{1p}/h_{pp} & & \\ & \ddots & & \vdots & & \\ & & 1 & -h_{p-1,p}/h_{pp} & & \\ & & & 1/h_{pp} & & \\ & & & -h_{p+1,p}/h_{pp} & 1 & \\ & & & \vdots & & \ddots \\ & & & -h_{mp}/h_{pp} & & 1 \end{pmatrix}. \tag{2.8}$$

(Таким образом, матрица $\overline{B}$ будет невырожденной только в том случае, если диагональный элемент h_{pp} матрицы H отличен от нуля.)

Примечание. Рассмотренные формулы можно использовать для достаточно простой реализации факторизации в виде так называемого *«эта-файла»* [4, 6]. В этой реализации текущая базисная матрица B хранится в следующем факторизованном виде:

$$B = B_0 H_1 \dots H_k, \tag{2.9}$$

где B_0 — начальная базисная матрица (т. е. после выполнения операции вычисления факторизации $B = B_0$), представленная в подходящем факторизованном виде (например, в виде обычного треугольного разложения), позволяющем эффективно решать системы вида $B_0 x = y$ и $B_0^T x = y$; $H_1, \dots, H_k$ — матрицы вида (2.3), отражающие замену столбцов в B_0 и хранимые в виде последовательного файла (массива), называемого «эта-файл» (так как буква H похожа на греческую «эта»). Из (2.9) следует, что:

$$x = B^{-1} y = H_k^{-1} \dots H_1^{-1} B_0^{-1} y, \tag{2.10}$$

т. е. операцию FTRAN можно выполнить, решая вначале систему с матрицей B_0 и затем последовательно с матрицами $H_1, \dots, H_k$ («эта-файл» просматривается в прямом порядке, откуда название операции — прямое преобразование). Транспонируя обе части формулы (2.10), получим:

$$x = B^{-T} y = B_0^{-T} H_1^{-T} \dots H_k^{-T} y, \tag{2.11}$$

т. е. операцию BTRAN можно выполнить, решая последовательно системы с матрицами $H_k^T, \dots, H_1^T$ и затем с матрицей B_0^T («эта-файл» просматривается в обратном порядке, откуда название операции — обратное преобразование). Наконец, операция замены столбца представляет собой вычисление и добавление очередной матрицы H_{k+1} к факторизации (2.9), т. е. в конец «эта-файла». (Отметим, что для факторизации B_0 также можно использовать «эта-файл», поскольку любую невырожденную матрицу можно получить, последовательно заменяя столбцы единичной матрицы.)

2.2 Основные операции

Анализ принципиальной схемы симплекс-метода (см. подразд. 1.3) позволяет выделить следующие основные операции, которые необходимы при решении ЛП-задачи (1.1)—(1.3) модифицированным симплекс-методом.

1. *Вычисление значений базисных переменных* $\beta = (\beta_i)$. Вектор β необходим для определения величины $\theta_{\min}$ (1.23), т. е. для выбора базисной переменной, которую следует исключить из числа базисных. Кроме того, этот вектор вместе с вектором активных границ (т. е. текущих значений) небазисных переменных f определяет текущее допустимое базисное решение (1.11). Из (1.12) следует, что вектор β можно вычислить, используя операцию FTRAN (см. подразд. 2.1), по формуле:

$$\beta = B^{-1}(b - Nf), \tag{2.12}$$

где b — вектор правых частей ограничений (1.2), N — матрица (1.6), составленная из столбцов матрицы коэффициентов ограничений A, соответствующих небазисным переменным x_N.

2. *Вычисление симплексных множителей* $\pi = (\pi_i)$. Вектор π необходим для вычисления относительных оценок небазисных переменных. Этот вектор можно вычислить, используя операцию BTRAN (см. подразд. 2.1), по формуле (1.16):

$$\pi = B^{-T}c_B, \tag{2.13}$$

где c_B — вектор коэффициентов целевой функции при базисных переменных x_B (1.4).

3. *Вычисление относительных оценок* $d = (d_j)$. Вектор d необходим для выбора небазисной переменной, которую следует включить в число базисных. Этот вектор можно вычислить по формуле (1.15):

$$d = c_N - N^T\pi, \tag{2.14}$$

c_N — вектор коэффициентов целевой функции при небазисных переменных x_N (1.4). Заметим, что указанная формула позволяет эффективно вычислять относительные оценки отдельных небазисных переменных. Пусть $(x_N)_j = x_k$ — некоторая небазисная переменная. Тогда ее относительная оценка есть $d_j = e_j^T d$, где e_j — j-й столбец единичной матрицы. Поэтому:

$$d_j = e_j^T c_N - e_j^T N^T \pi = (c_N)_j - N_j^T \pi = c_k - A_k^T \pi, \tag{2.15}$$

где $(c_N)_j = c_k$ — коэффициент целевой функции и $N_j = A_k$ — столбец матрицы коэффициентов ограничений, соответствующие небазисной переменной $(x_N)_j = x_k$.

4. *Вычисление столбца симплекс-таблицы* $\Xi_q = (\xi_{iq})$, который соответствует выбранной небазисной переменной $(x_N)_q$. Вектор-столбец Ξ_q совместно с вектором текущих значений базисных переменных β необходим для определения величины $\theta_{\min}$ (см. выше). Так как $\Xi_q = \Xi e_q$, где e_q — q-й столбец единичной матрицы, то из (1.9) следует, что столбец Ξ_q можно вычислить, используя операцию FTRAN, по формуле:

$$\Xi_q = -B^{-1}Ne_q = -B^{-1}N_q, \tag{2.16}$$

где N_q — столбец матрицы коэффициентов ограничений A, соответствующий небазисной переменной $(x_N)_q$.

5. *Переход к смежному базису.* Эта операция состоит в замене столбца B_p текущей базисной матрицы B столбцом N_q текущей матрицы N, где индексы p и q определяются выбранными базисной $(x_B)_p$ и небазисной $(x_N)_q$ переменными, в результате чего получается новая базисная матрица $\overline{B}$, соответствующая смежному базису. Поскольку базисная матрица не хранится явно, а используется ее факторизация, то выполнение данной операции сводится к пересчету факторизации базисной матрицы (см. подразд. 2.1).

Все перечисленные операции допускают эффективное выполнение, если матрица коэффициентов ограничений A доступна по столбцам.[14]

В заключение отметим, что если придерживаться принципиальной схемы симплекс-метода из подразд. 1.3, то каждая итерация модифицированного симплекс-метода требует выполнения двух операций FTRAN, одной операции BTRAN, и одного пересчета факторизации текущей базисной матрицы после замены столбца.

[14]Различие между строчным и столбцовым представлениями матрицы является существенным только если для хранения матрицы используется разреженный формат.

2.3 Пересчет компонент базисного решения

При решении ЛП-задач большой размерности число итераций симплекс-метода может достигать десятков и сотен тысяч. Поэтому для повышения вычислительной эффективности модифицированного симплекс-метода на каждой итерации вместо прямого вычисления основных компонент базисного решения по формулам (2.12), (2.13) и (2.14) выгоднее выполнять их пересчет, т. е. использовать известные значения этих компонент в текущем базисе для вычисления их значений в смежном базисе. Здесь необходимо отметить, что подобный пересчет неизбежно сопровождается накоплением ошибок округления, поэтому время от времени следует использовать прямое вычисление этих компонент (например, всякий раз после начального вычисления факторизации базисной матрицы).

Пересчет значений базисных переменных $\beta = (\beta_i)$. Пусть переменная $(x_B)_p$ выходит из базиса на свою активную границу $t = (l_B)_p$ или $t = (u_B)_p$, а переменная $(x_N)_q$ входит в базис. Так как остальные небазисные переменные не изменяются, то вектор приращений небазисных переменных при переходе к смежному базису есть

$$\Delta x_N = \Delta(x_N)_q e_q, \tag{2.17}$$

где $\Delta(x_N)_q$ — приращение переменной $(x_N)_q$, e_q — q-й столбец единичной матрицы. Из соотношения (1.8) следует, что в этом случае вектор приращений базисных переменных равен

$$\Delta x_B = \Xi \Delta x_N = \Delta(x_N)_q \Xi e_q = \Delta(x_N)_q \Xi_q, \tag{2.18}$$

где $\Xi_q = (\xi_{iq})$ — q-й столбец текущей симплекс-таблицы. В частности, для отдельной базисной переменной имеем:

$$\Delta(x_B)_i = \xi_{iq} \Delta(x_N)_q. \tag{2.19}$$

Зная приращение переменной $(x_B)_p$, которое, очевидно, равно

$$\Delta(x_B)_p = t - \beta_p, \tag{2.20}$$

где β_p — значение этой переменной в текущем базисе, из формулы (2.19) при $i = p$ следует, что соответствующее приращение переменной $(x_N)_q$ равно[15]

$$\Delta(x_N)_q = \frac{\Delta(x_B)_p}{\xi_{pq}} = \frac{t - \beta_p}{\xi_{pq}}. \tag{2.21}$$

[15]Понятно, что если выбор переменной $(x_B)_p$ выполнялся на основе величины $\theta_{\min}$ (см. принципиальную схему симплекс-метода, подразд. 1.3), то $\Delta(x_N)_q = s\theta_{\min}$.

А так как в смежном базисе $(x_N)_q$ занимает место $(x_B)_p$, то

$$\overline{\beta}_p = f_q + \Delta(x_N)_q, \tag{2.22}$$

где f_q — активная граница переменной $(x_N)_q$ в текущем базисе. Остальные базисные переменные остаются базисными в смежном базисе, поэтому из (2.19) следует, что

$$\overline{\beta}_i = \beta_i + \Delta(x_B)_i = \beta_i + \xi_{iq}\Delta(x_N)_q,\ i \neq p. \tag{2.23}$$

В специальном случае, когда в смежном базисе переменная $(x_N)_q$ переходит со своей текущей активной границы на противоположную, для определения приращения этой переменной вместо формулы (2.21) следует использовать формулу

$$\Delta(x_N)_q = s[(u_N)_q - (l_N)q], \tag{2.24}$$

где $s = +1$, если $(x_N)_q$ возрастает, или $s = -1$, если $(x_N)_q$ убывает. При этом значения *всех* базисных переменных пересчитываются по формуле (2.23).

Использование формул пересчета (2.22) и (2.23) позволяет сэкономить одну операцию FTRAN на каждой итерации модифицированного симплекс-метода, поскольку используемый в этих формулах столбец симплекс-таблицы Ξ_q уже будет вычислен ранее (он требуется для определения величины $\theta_{\min}$).

Пересчет относительных оценок $d = (d_j)$. Формулы пересчета относительных оценок небазисных переменных — это формулы (1.37) и (1.38) (см. подразд 1.4):

$$\overline{d}_q = \frac{d_q}{\xi_{pq}}, \tag{2.25}$$

$$\overline{d}_j = d_j - \xi_{pj}\overline{d}_q,\ j \neq q. \tag{2.26}$$

Заметим, что в специальном случае, когда в смежном базисе небазисная переменная $(x_N)_q$ переходит со своей текущей активной границы на противоположную, *все* относительные оценки, включая оценку d_q, остаются *неизменными*.

Примечание. Так как пересчет сопровождается ошибками округления, то перед вычислением $\overline{d}_q$ по формуле (2.25) целесообразно использовать уже вычисленный столбец Ξ_q текущей симплекс-таблицы для прямого, а значит, более точного вычисления d_q в текущем базисе. Поскольку $d_q = e_q^T d$, где e_q — q-й столбец единичной матрицы, то необходимая формула получается умножением слева e_q^T на обе части равенства (1.15):

$$d_q = (c_N)_q + \Xi_q^T c_B. \tag{2.27}$$

Для пересчета оценок по указанным формулам требуется p-я строка симплекс-таблицы $\xi_p = (\xi_{pj})$ в текущем базисе. Так как $\xi_p = \Xi^T e_p$, где e_p — p-й столбец единичной матрицы, то из (1.9) следует, что вектор ξ_p можно вычислить по формуле:

$$\xi_p = -N^T B^{-T} e_p = -N^T \rho, \tag{2.28}$$

где вектор ρ есть p-я строка обратной матрицы B^{-1}, который можно вычислить, используя операцию BTRAN:

$$\rho = B^{-T} e_p. \tag{2.29}$$

Сравнивая формулы (2.28) и (2.29) с формулами прямого вычисления оценок (2.13) и (2.14) можно заметить, что они имеют идентичную структуру,[16] поэтому на первый взгляд пересчет относительных оценок не позволяет повысить эффективность вычислений по сравнению с их прямым вычислением. Однако, в отличие от вектора c_B в формуле (2.13), вектор e_p является единичным вектором, поэтому можно ожидать, что вектор ρ будет более разреженным, чем вектор π. Это, во-первых, может увеличить эффективность выполнения операции BTRAN (что, конечно, зависит от используемой факторизации базисной матрицы). И во-вторых, разреженность вектора ρ можно использовать для более эффективного вычисления строки ξ_p. Последнее обстоятельство обусловлено тем, что произведение матрицы на вектор можно вычислять двумя способами: как скалярные произведения строк матрицы на заданный вектор и как линейную комбинацию столбцов матрицы. Первый способ не позволяет учесть разреженность заданного вектора, но при использовании второго способа можно пропускать те столбцы матрицы, которым соответствуют нулевые элементы вектора. Применительно к формуле (2.28), где столбцами N^T являются строки матрицы N, второй способ означает, что вектор ξ_p вычисляется как линейная комбинация строк матрицы N:

$$\xi_p = -N^T \rho = -\sum_{i=1}^{m} \rho_i \, (i\text{-я строка матрицы } N). \tag{2.30}$$

Эффективное использование формулы (2.30) возможно лишь в том случае, когда матрица N доступна по строкам. Поэтому возникает техническая проблема, так как стандартный вариант модифицированного симплекс-метода предполагает хранение в явном виде лишь матрицы коэффициентов ограничений A в разреженном столбцовом формате (а также базисной матрицы B в факторизованном виде).

[16] Это связано с тем, что p-ю строку симплекс-таблицы можно рассматривать как строку относительных оценок небазисных переменных для целевой функции, где коэффициент при базисной переменной $(x_B)_p$ равен единице, а коэффициенты при всех остальных переменных равны нулю.

Для решения указанной проблемы Р. Биксби (R. E. Bixby) [7] предлагает хранить матрицу N в разреженном строчном формате. Заметим, что матрица N изменяется на каждой итерации, поэтому всякий раз после перехода к смежному базису старый столбец N_q, соответствующий выбранной небазисной переменной $(x_N)_q$, следует заменять новым столбцом A_k, соответствующим выбранной базисной переменной $(x_B)_p = x_k$. Так как N хранится по строкам, то для этого сначала нужно пройти по ненулевым элементам старого столбца, который доступен как соответствующий столбец матрицы A, удалить эти элементы из строк N, после чего пройти по ненулевым элементам нового столбца A_k и добавить эти элементы в строки N. По всей видимости технику Биксби имеет смысл применять в тех случаях, когда большинство столбцов матрицы A являются достаточно разреженными (что является типичным для большинства реальных ЛП-задач).

В качестве компромиссного варианта вместо хранения матрицы N можно хранить дубликат матрицы A в разреженном строчном формате. Это дает преимущество по сравнению с техникой Биксби, поскольку матрица A остается неизменной при переходе к смежному базису. Однако в этом случае непосредственное использование формулы (2.30) оказывается невозможным. Поэтому вместо вычисления вектора ξ_p как линейной комбинации строк матрицы N предлагается вычислять вектор α_p как линейную комбинацию строк матрицы A:

$$\alpha_p = -A^T\rho = -\sum_{i=1}^{m} \rho_i \,(i\text{-я строка матрицы } A). \tag{2.31}$$

Поскольку столбцы N являются подмножеством столбцов A, то используя соотношение (1.6) требуемый вектор ξ_p можно собрать из соответствующих элементов вектора α_p. Заметим, что при вычислении линейной комбинации α_p вычисляются «лишние» элементы (соответствующие столбцам матрицы B),[17] поэтому эффективность формулы (2.31) снижается с ростом числа ненулевых элементов вектора ρ. В связи с этим использовать формулу (2.31) предлагается лишь в тех случаях, когда заполнение ρ не очень велико (скажем, не более 10%), а в остальных случаях вычислять элементы вектора ξ_p как обычные скалярные произведения столбцов N на вектор ρ.

[17] Этими «лишними» элементами будут, очевидно, элементы единичного вектора e_p, так как из формулы (2.29) следует, что $B^T\rho = e_p$.

3 Выбор небазисной переменной

В соответствии с принципиальной схемой симплекс-метода, рассмотренной в конце подразд. 1.3, на шаге 2 требуется выбрать подходящую небазисную переменную $(x_N)_q$, изменение которой в допустимом направлении приводит к улучшению (уменьшению) целевой функции.[18] С геометрической точки зрения выбор небазисной переменной соответствует выбору ребра многогранного множества допустимых решений ЛП-задачи, исходящего из текущей вершины, вдоль которого имеет место улучшение целевой функции.

Пусть $J \subseteq \{1, \ldots, n-m\}$ — множество индексов небазисных переменных $(x_N)_j$, относительная оценка d_j которых имеет «неправильный» знак. Если $J \neq \varnothing$, то текущее базисное решение не является оптимальным, и в этом случае в качестве небазисной переменной можно, в принципе, выбрать любую переменную из J. Таким образом, проблема выбора состоит в принятии решения, какую именно небазисную переменную следует выбирать на каждой итерации симплекс-метода.

В данном разделе рассмотрены основные стратегии выбора небазисной переменной, получившие практическое применение.

3.1 Правило Данцига

Правило Данцига [2] состоит в выборе подходящей небазисной переменной с максимальной по абсолютной величине относительной оценкой, т. е. переменной $(x_N)_q$, для которой

$$|d_q| = \max_{j \in J} |d_j|. \tag{3.1}$$

Поскольку при изменении небазисной переменной $(x_N)_j$ на величину s целевая функция изменяется на величину $d_j s$, то правило Данцига обеспечивает наибольшее относительное улучшение целевой функции.

Основным преимуществом правила Данцига является его простота, поэтому оно является традиционным для табличного симплекс-метода. Однако фактическое изменение целевой функции при переходе к смежному базису определяется не только относительной оценкой выбранной небазисной переменной, но и фактическим изменением этой переменной, т. е. величиной $\theta_{\min}$ (1.23), которая зависит от выбранной небазисной переменной. При этом, как показывает практика, использование правила Данцига при решении реальных ЛП-задач во многих случях приводит к существенно бóльшему числу итераций симплекс-метода, необходимых для достижения оптимальной точки, по сравнению с другими, менее тривиальными стратегиями выбора небазисной переемнной.

[18] Поскольку небазисные переменные соответствуют столбцам текущей симплекс-таблицы (1.8), то данную операцию часто называют *выбором столбца*.

3.2 Метод наиболее крутого ребра

В своей статье [8] П. Харрис (P. M. J. Harris) заметила, что возможной причиной слишком медленной сходимости симплекс-метода при использовании правила Данцига является его асимметрия. Действительно, это правило основано на использовании относительных оценок d_j. Однако оценка, которая является максимальной по абсолютной величине с точки зрения текущего базиса уже может не быть таковой с точки зрения смежного базиса, поскольку симплекс-таблица для смежного базиса получается в результате линейного преобразования,[19] которое не является ортогональным, а значит, не сохраняет расстояния и углы.

Чтобы устранить указанную асимметрию, Д. Гольдфарб (D. Goldfarb) и Дж. Рид (J. K. Reid) предложили использовать *метод наиболее крутого ребра (steepest edge)* [9]. В соответствии с этим методом предлагается выбирать ту подходящую небазисную переменную, для которой направление, определяемое соответствующим ребром, образует наиболее острый угол с направлением антиградиента (в случае минимизации) или градиента (в случае максимизации) целевой функции.

Отличительная особенность метода наиболее крутого ребра состоит в том, что указанный угол измеряется в пространстве всех (базисных и небазисных) переменных, а значит, не зависит от текущего базиса, поскольку пространство всех переменных остается одним и тем же на протяжении всех итераций. В этом смысле правило Данцига соответствует выбору ребра многогранника, которое также образует наиболее острый угол с антиградиентом (или градиентом в случае максимизации) целевой функции, но только в *текущем подпространстве небазисных переменных*, которое зависит от текущего базиса и, следовательно, изменяется на каждой итерации, что приводит к появлению асимметрии.

Рассмотрим вывод соответствующих формул.

Опуская перестановочную матрицу Π в формуле (1.11) запишем текущее базисное решение в виде:

$$x^0 = (x_B; x_N) = [(x_B)_1, \ldots, (x_B)_m; (x_N)_1, \ldots, (x_N)_{n-m}].$$

Геометрически точка x^0 является текущей вершиной многогранного множества в пространстве *всех* переменных ЛП-задачи. Допустим теперь, что некоторая небазисная переменная $(x_N)_j$ получила единичное приращение, причем все остальные небазисные переменные остались неизменными. Из (1.8) следует, что в результате получится следующая точка:

$$x^j = [(x_B)_1 + \xi_{1j}, \ldots, (x_B)_m + \xi_{mj}; (x_N)_1, \ldots, (x_N)_j + 1, \ldots, (x_N)_{n-m}],$$

где $\xi_{1j}, \ldots, \xi_{mj}$ — элементы j-го столбца текущей симплекс-таблицы Ξ. Поскольку указанная смещенная точка лежит на прямой, проходящей

[19] Это преобразование соответствует одному шагу исключения метода Гаусса—Жордана.

через ребро многогранного множества, соответствующего небазисной переменной $(x_N)_j$, то направление этого ребра совпадает с направлением вектора-приращения:

$$\Delta x^j = x^j - x^0 = (\xi_{1j}, \ldots, \xi_{mj}; 0, \ldots, 1, \ldots, 0), \tag{3.2}$$

где единица расположена в позиции $(x_N)_j$.

Нас интересует угол φ_j между направлением ребра многогранного множества Δx_j и градиентом целевой функции ∇z:

$$\cos\varphi_j = \frac{(\Delta x^j)^T \nabla z}{\|\Delta x^j\| \|\nabla z\|}. \tag{3.3}$$

Для целевой функции имеем $z = c^T x = c_B x_B + c_N x_N$, откуда:

$$\nabla z = c = [(c_B)_1, \ldots, (c_B)_m; (c_N)_1, \ldots, (c_N)_{n-m}]. \tag{3.4}$$

Заметим, что из формул (1.15) и (1.16) с учетом определения (1.9) следует $d = c_N - N^T B^{-T} c_B = \Xi^T c_B + c_N$. Поэтому, используя формулы (3.3) и (3.4), имеем:

$$(\Delta x^j)^T \nabla z = \sum_{i=1}^{m} \xi_{ij}(c_B)_i + (c_N)_j = \Xi_j^T c_B + (c_N)_j = d_j, \tag{3.5}$$

где Ξ_j — j-й столбец текущей симплекс-таблицы, d_j — относительная оценка небазисной переменной $(x_N)_j$. Подставляя далее выражение для скалярного произведения из (3.5) в (3.3) окончательно получим:

$$\cos\varphi_j = \frac{d_j}{\|\Delta x^j\| \|\nabla z\|}, \tag{3.6}$$

где

$$\|\Delta x^j\| = \left(\sum_{i=1}^{m} \xi_{ij}^2 + 1 \right)^{1/2} = (\Xi_j^T \Xi_j + 1)^{1/2}, \tag{3.7}$$

$$\|\nabla z\| = \|c\| = (c^T c)^{1/2}. \tag{3.8}$$

Поскольку градиент целевой функции $\nabla z = c = \text{const}$ не зависит от текущего базиса, то для применения метода наиболее крутого ребра достаточно знать только относительные оценки d_j и нормы векторов-приращений $\|\Delta x^j\|$. При этом выбор наиболее подходящей небазисной переменной $(x_N)_q$ определяется следующим правилом (ср. с правилом Данцига из подразд 3.1):

$$\frac{|d_q|}{\|\Delta x^q\|} = \max_{j \in J} \frac{|d_j|}{\|\Delta x^j\|} = \max_{j \in J} \frac{|d_j|}{(\Xi_j^T \Xi_j + 1)^{1/2}}. \tag{3.9}$$

В случае табличного симплекс-метода вычисление норм $\|\Delta x^j\|$ не вызывает затруднений, поскольку симплекс-таблица $\Xi = \xi_{ij}$ доступна в явном виде на каждой итерации. Однако, если используется модифицированный симплекс-метод, вычисление отдельного столбца Ξ_j, необходимого для вычисления $\|\Delta x^j\|$, требует применения операции прямого преобразования (FTRAN). Поэтому прямое вычисление указанных норм для всех небазисных переменных хотя и возможно, но является неоправданно трудоемким, из-за чего такой способ не находит практического применения.

Чтобы избавиться от необходимости явного вычисления норм векторов-приращений для всех небазисных переменных на каждой итерации симплекс-метода, в работе [9] были предложены формулы пересчета этих норм при переходе к смежному базису (по аналогии с пересчетом компонент базисного решения из подразд. 2.3). Рассмотрим вывод этих формул.

Введем для удобства величины:

$$\gamma_j = \|\Delta x^j\|^2 = 1 + \Xi_j^T \Xi_j = 1 + \sum_{i=1}^{m} \xi_{ij}^2,\ j = 1, \ldots, n-m, \tag{3.10}$$

и допустим, что для текущего базиса эти величины уже известны. Допустим также, что в смежном базисе переменная $(x_B)_p$ становится небазисной, а $(x_N)_q$ — базисной, т. е. ведущим элементом симплекс-таблицы является ξ_{pq}. Нас интересуют формулы пересчета величин γ_j для смежного базиса.

Рассмотрим вначале пересчет величины γ_q для ведущего столбца ($j = q$). Используя формулы пересчета элементов симплекс-таблицы (1.33)—(1.36) из подразд. 1.4, имеем:

$$\overline{\gamma_q} = 1 + \sum_i \overline{\xi}_{iq}^2 = 1 + \overline{\xi}_{pq}^2 + \sum_{i \neq p} \overline{\xi}_{iq}^2 = 1 + \left(\frac{1}{\xi_{pq}}\right)^2 + \sum_{i \neq p} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2.$$

Заметим, что:

$$\sum_i \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = \left(\frac{\xi_{pq}}{\xi_{pq}}\right)^2 + \sum_{i \neq p} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = 1 + \sum_{i \neq p} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2,$$

поэтому:

$$\overline{\gamma}_q = \left(\frac{1}{\xi_{pq}}\right)^2 + \sum_i \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = \frac{1}{\xi_{pq}^2}\left(1 + \sum_i \xi_{iq}^2\right),$$

откуда окончательно следует, что:

$$\overline{\gamma}_q = \frac{1}{\xi_{pq}^2}\gamma_q. \tag{3.11}$$

Рассмотрим теперь пересчет величин γ_j для остальных столбцов ($j \neq q$). Снова используя формулы пересчета элементов симплекс-таблицы (1.33)—(1.36) из подразд. 1.4, имеем:

$$\overline{\gamma}_j = 1 + \sum_i \overline{\xi}_{ij}^2 = 1 + \overline{\xi}_{pj}^2 + \sum_{i \neq p} \overline{\xi}_{ij}^2 = 1 + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_{i \neq p} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2.$$

Заметим, что:

$$\sum_i \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 = \sum_{i \neq p} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 + \left(\xi_{pj} - \frac{\xi_{pq}\xi_{pj}}{\xi_{pq}}\right)^2 =$$
$$= \sum_{i \neq p} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2,$$

поэтому:

$$\overline{\gamma}_j = 1 + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_i \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 =$$
$$= 1 + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_i \xi_{ij}^2 + \sum_i \left(\frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 - 2\sum_i \frac{\xi_{ij}\xi_{iq}\xi_{pj}}{\xi_{pq}} =$$
$$= \left(1 + \sum_i \xi_{ij}^2\right) + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \left(1 + \sum_i \xi_{iq}^2\right) - 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) \sum_i \xi_{ij}\xi_{iq}.$$

откуда:

$$\overline{\gamma}_j = \gamma_j + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \gamma_q - 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) \sum_i \xi_{ij}\xi_{pq}.$$

Сумма $\sum_i \xi_{ij}\xi_{pq}$ представляет собой скалярное произведение j-го столбца симплекс-таблицы:

$$\Xi_j = \Xi e_j = B^{-1}Ne_j = -B^{-1}N_j$$

и ее q-го (ведущего) столбца Ξ_q, поэтому:

$$\sum_i \xi_{ij}\xi_{pq} = \Xi_j^T \Xi_q = (-B^{-1}N_j)\Xi_q = -N_j^T B^{-T} \Xi_q.$$

Таким образом, окончательно получаем:

$$\overline{\gamma}_j = \gamma_j + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \gamma_q + 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) N_j^T B^{-T} \Xi_q. \tag{3.12}$$

Использование формулы пересчета (3.12) требует двух операций обратного преобразования (BTRAN) на каждой итерации модифицированного симплекс-метода: одна операция нужна для вычисления элементов ведущей строки симплекс-таблицы $\xi_p = (\xi_{pj})$,[20] а вторая — для вычисления вектора $B^{-T}\Xi_q$.

Следует отметить, что многократный пересчет величин γ_j вследствие ошибок округления может привести к значительному отклонению этих величин от их точных значений. Чтобы уменьшить влияние ошибок округления, в работе [3] предлагается, во-первых, использовать в формуле (3.11) не текущее значение γ_q с предыдущей итерации, а более точное значение этой величины, вычисленное непосредственно по формуле (3.10):

$$\gamma_q = 1 + \sum_i \xi_{iq}^2, \tag{3.13}$$

что не вызывает затруднений, поскольку в случае модифицированного симплекс-метода ведущий столбец текущей симплекс-таблицы $\Xi_q = (\xi_{iq})$ доступен в явном виде. Во-вторых, можно заметить, что:

$$\overline{\gamma}_j = 1 + \sum_i \overline{\xi}_{ij}^2 = 1 + \overline{\xi}_{pj}^2 + \sum_{i \neq p} \overline{\xi}_{ij}^2 \geq 1 + \overline{\xi}_{pj}^2 = 1 + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2. \tag{3.14}$$

Поэтому, если вследствие ошибок округления значение $\overline{\gamma}_q$, вычисленное по формуле (3.12), оказывается меньше, чем правая часть неравенства (3.14), то в качестве более точного значения $\overline{\gamma}_q$ предлагается использовать величину в правой части указанного неравенства.

Допустим, что p-я (ведущая) строка $\xi_p = (\xi_{pj})$ и q-й (ведущий) столбец $\Xi_q = (\xi_{iq})$ текущей симплекс-таблицы уже вычислены. Тогда практическая схема пересчета величин γ_j для смежного базиса может быть следующей.

1. Вычислить более точное значение γ_q для текущего базиса:

$$\gamma_q = 1 + \sum_{i=1}^{m} \xi_{iq}^2.$$

2. Вычислить вспомогательный вектор, используя операцию BTRAN:

$$u = B^{-T}\Xi_q.$$

3. Выполнить пп. 4—6 для $j = 1, \ldots, q-1, q+1, \ldots, n-m$.

4. Вычислить вспомогательную величину:

$$r_j = \xi_{pj}/\xi_{pq}.$$

[20] Эта строка также используется для пересчета относительных оценок небазисных переменных (см. подразд. 2.3).

5. Вычислить вспомогательное скалярное произведение:

$$s_j = N_j^T u,$$

где N_j — j-й столбец матрицы N, т. е. столбец матрицы коэффициентов ограничений A, соответствующий небазисной переменной $(x_N)_j$.

6. Вычислить величину $\overline{\gamma}_j$ для смежного базиса:

$$\overline{\gamma}_j = \max(\gamma_j + r_j^2\gamma_q + 2r_j s_j,\ 1 + r_j^2).$$

7. Вычислить величину $\overline{\gamma}_q$ для смежного базиса:

$$\overline{\gamma}_q = \gamma_q/\xi_{pq}^2.$$ ■

Последнее замечание касается использования правила (3.9). Чтобы избежать извлечения квадратного корня для выбора небазисной переменной $(x_N)_q$ по методу наиболее крутого ребра, можно использовать эквивалентное правило:

$$\frac{d_q^2}{\gamma_q} = \max_{j \in J} \frac{d_j^2}{\gamma_j}. \tag{3.15}$$

Практическое решение задач линейного программирования с использованием метода наиболее крутого ребра в большинстве случаев приводит к существенному сокращению числа итераций симплекс-метода по сравнению со стандартным правилом выбора небазисных переменных, и как следствие этого, к уменьшению общего времени решения даже с учетом того, что на каждой итерации приходиться выполнять дополнительные вычисления.

Однако метод наиболее крутого ребра имеет один существенный практический недостаток. Как уже было отмечено выше, многократный пересчет величин γ_j на каждой итерации симплекс-метода может привести к значительному отклонению этих величин от их точных значений вследствие ошибок округления. Поэтому в какой-то момент может возникнуть необходимость прямого вычисления этих величин для текущего базиса непосредственно по формуле (3.10), для чего требуется вычислить *все* столбцы текущей симплекс-таблицы. С практической точки зрения это является неприемлемым, так как в случае модифицированного симплекс-метода вычисление одного столбца симплекс-таблицы требует выполнения одной операции прямого преобразования (FTRAN).

Аналогичная ситуация возникает и перед выполнением самой первой итерации симплекс-метода, когда величины γ_j еще не известны. Использование формулы (3.10) не вызывает затруднений лишь в единственном случае, а именно, когда начальный базис полностью состоит из вспомогательных переменных ограничений[21] (т. е. когда все исходные структурные переменные являются небазисными), поскольку в этом случае

[21] Введение вспомогательных неотрицательных или неположительных переменных в ограничения задачи является стандартной техникой преобразования неравенств к виду равенств.

$B = I$, так что $\Xi = -B^{-1}N = -N$. Однако стандартный начальный базис является в некотором смысле исключением, поскольку все современные пакеты линейного программирования используют те или иные процедуры для формирования менее тривиального начального базиса. То же самое относится и к повторной оптимизации, например, при использовании техники генерации столбцов, где начальным базисом служит оптимальный базис редуцированной задачи.

3.3 Проекционный метод наиболее крутого ребра

Проекционный метод наиболее крутого ребра является развитием метода наиболее крутого ребра, рассмотренного в предыдущем подразделе. Основная идея этого метода состоит в том, чтобы измерять углы φ_j (3.3) не в пространстве всех переменных E^n а в некотором подпространстве $V \subseteq E^n$, называемом *эталонным*, которое также остается неизменным на протяжении достаточно большого числа итераций симплекс-метода. Другими словами, в этом методе вместо формулы (3.3) используется формула:

$$\cos\varphi_j = \frac{(\Delta x_V^j)^T \nabla z_V}{\|\Delta x_V^j\| \|\nabla z_V\|}, \tag{3.16}$$

где Δx_V^j и ∇z_V — проекции, соответственно, векторов Δx^j и ∇z на эталонное подпространство V. При этом с практической точки зрения наиболее подходящим эталонным подпространством является *координатное подпространство*, определяемое небазисными переменными на некоторой произвольно выбранной итерации симплекс-метода.

Примечание. Правомочность выбора подпространства небазисных переменных в качестве эталонного имеет следующее обоснование. При рассмотрении табличного симплекс-метода (см. подразд. 1.4) было отмечено, что преобразованная задача (1.25)—(1.28), имеющая место на отдельной итерации симплекс-метода, эквивалентна исходной ЛП-задаче (1.1)—(1.3). Если исключить базисные переменные из преобразованной задачи, то мы получим следующую *редуцированную* задачу:

$$z = d^T x_N + d_0 \to \min$$

$$l_B \le \Xi x_N + g \le u_B$$

$$l_N \le x_N \le u_N$$

Эта редуцированная задача, очевидно, также эквивалентна исходной задаче, но включает только небазисные переменные исходной задачи, которые можно рассматривать как обычные переменные редуцированной задачи. Таким образом, пространство всех переменных редуцированной задачи — это подпространство небазисных переменных исходной задачи, выбранных на соответствующей итерации симплекс-метода. Следовательно, метод наиболее крутого ребра в пространстве всех переменных применительно к редуцированной задаче эквивалентен проекционному методу наиболее крутого ребра в подпространстве небазисных переменных применительно к исходной задаче.

Рассмотрим вывод соответствующих формул, полагая, что вектор-приращение Δx^j имеет вид (3.2).

Пусть V — эталонное координатное подпространство переменных, которое первоначально (т. е. на некоторой произвольно выбранной итерации симплекс-метода) совпадает с подпространством небазисных переменных. Определим следующие вспомогательные величины для текущего базиса:

$$\eta_i = \begin{cases} 1, & \text{если } (x_B)_i \in V \\ 0, & \text{если } (x_B)_i \notin V \end{cases} \tag{3.17}$$

$$\delta_j = \begin{cases} 1, & \text{если } (x_N)_j \in V \\ 0, & \text{если } (x_N)_j \notin V \end{cases} \tag{3.18}$$

Опуская перестановочную матрицу Π в формуле (1.11), можно считать, что $\Delta x^j = (\Delta x^j_B; \Delta x^j_N)$ и $\nabla z = (\nabla z_B; \nabla z_N)$, где подвекторы Δx^j_B и ∇z_B соответствуют базисным переменным x_B, а подвекторы Δx^j_N и ∇z_N соответствуют небазисным переменным x_N. Поэтому проекции указанных векторов на подпространство V будут равны:

$$\Delta x^j_V = (\eta \Delta x^j_B; \delta \Delta x^j_N), \tag{3.19}$$

$$\nabla z_V = (\eta \nabla z_B; \delta \nabla z_N), \tag{3.20}$$

где $\eta = \operatorname{diag}(\eta_i)$, $\delta = \operatorname{diag}(\delta_j)$ — диагональные (0,1)-матрицы. (Другими словами, проекция вектора на подпространство V получается в результате замены нулями тех его координат, которые не входят в V.)

Обратимся вначале к скалярному произведению в числителе формулы (3.16). Используя соотношения (3.19) и (3.20), а также учитывая, что $\eta^T \eta = \eta$ и $\delta^T \delta = \delta$, получим:

$$\begin{aligned} (\Delta x^j_V)^T \nabla z_V &= (\Delta x^j_B)^T \eta^T \eta \nabla z_B + (\Delta x^j_N)^T \delta^T \delta \nabla z_B = \\ &= (\Delta x^j_B)^T \eta \nabla z_B + (\Delta x^j_N)^T \delta \nabla z_B = (\Delta x^j)^T \nabla z_V. \end{aligned} \tag{3.21}$$

В любом базисе исходная целевая функция (1.1) эквивалентна целевой функции преобразованной задачи (1.25).[22] Следовательно, при использовании формулы (3.16) мы вправе считать, что начиная с той итерации симплекс-метода, когда выполняется фиксация эталонного подпространства V, мы вместо исходной целевой функции $z = c^T x + c_0$ имеем дело с *эквивалентной* целевой функцией преобразованной задачи $z = d^T x_N + d_0$ для указанной начальной итерации, как если бы

[22] Целевую функцию преобразованной задачи (1.25) можно получить вычитанием из исходной целевой функции (1.1) линейной комбинации ограничений-равенств (1.2), если в качестве коэффициентов этой линейной комбинации использовать симплексные множители π_i (1.16). Заметим, что коэффициенты отдельного ограничения-равенства определяют вектор, ортогональный к плоскому множеству (аффинному подпространству) точек, удовлетворяющих системе ограничений-равенств (1.2), поэтому линейная комбинация таких векторов также будет обладать этим свойством ортогональности. Отсюда следует, что градиенты эквивалентных целевых функций отличаются друг от друга лишь этой ортогональной составляющей.

исходной была эта эквивалентная целевая функция. Поскольку же на этой начальной итерации $\delta = \mathrm{diag}(\delta_j) = I$ (единичная матрица), то $\nabla z = \nabla z_V = (0; d)$, откуда с учетом (3.5) следует, что

$$(\Delta x^j)^T \nabla z_V = (\Delta x^j)^T \nabla z = d_j. \tag{3.22}$$

Важно отметить, что относительные оценки $d_j = (\Delta x^j)\nabla z$ для исходной целевой функции $z = c^T x + c_0$ и $d_j = (\Delta x^j)^T \nabla z_V$ для эквивалентной целевой функции $z = d^T x_N + d_0$ (где d и x_N относятся к итерации, на которой происходит фиксация эталонного подпространства) будут одними и теми же на всех последующих итерациях симплекс-метода вследствие эквивалентности целевых функций. Поэтому для вычисления относительных оценок d_j, которые относятся к эквивалентной целевой функции, никакая замена коэффициентов исходной целевой функции не требуется.

Подстановка скалярного произведения из (3.21) в числитель формулы (3.16) с учетом (3.22) дает следующее выражение для интересующего нас угла (подчеркнем, что теперь мы рассматриваем угол между проекцией вектора-приращения $\|\Delta x_V^j\|$ и проекцией градиента $\|\nabla z_V\|$ *эквивалентной* целевой функции, что однако не меняет существа дела):

$$\cos\varphi_j = \frac{d_j}{\|\Delta x_V^j\| \|\nabla z_V\|}. \tag{3.23}$$

Так как $\nabla z = \mathrm{const}$ и эталонное подпространство V не зависит от текущего базиса (остается неизменным), то $\nabla z_V = \mathrm{const}$.[23] Поэтому для проекционного метода наиболее крутого ребра выбор наиболее подходящей небазисной переменной $(x_N)_q$ определяется следующим правилом:

$$\frac{|d_q|}{\|\Delta x_V^q\|} = \max_{j \in J} \frac{|d_j|}{\|\Delta x_V^j\|}, \tag{3.24}$$

которое является аналогом правила (3.9) для обычного метода наиболее крутого ребра (см. подразд. 3.2).

Перейдем теперь к нормам проекций векторов-приращений на эталонное подпространство. Из формул (3.2) и (3.19) следует, что

$$\Delta x_V^j = (\eta_1 \xi_{1j}, \ldots, \eta_m \xi_{mj}; 0, \ldots, \delta_j, \ldots, 0), \tag{3.25}$$

откуда:

$$\|\Delta x_V^j\| = \left(\sum_{i=1}^{m} \eta_i \xi_{ij}^2 + \delta_j \right)^{1/2} = \left(\sum_{i \in R} \xi_{ij}^2 + \delta_j \right)^{1/2}, \tag{3.26}$$

[23] Очевидно, что $\nabla z = \nabla z_V$ в соответствии с определением данной эквивалентной целевой функции.

где $R = \{i : \eta_i = 1\} = \{i : (x_B)_i \in V\}$ — множество индексов базисных переменных, принадлежащих эталонному подпространству.

Введем для удобства величины:

$$\gamma_j = \|\Delta x_V^j\|^2 = \delta_j + \sum_{i \in R} \xi_{ij}^2,\ j = 1, \ldots, n - m, \tag{3.27}$$

которые аналогичны величинам (3.10) в обычном методе наиболее крутого ребра, и допустим, что для текущего базиса эти величины уже известны. Допустим также, что в смежном базисе переменная $(x_B)_p$ становится небазисной, а $(x_N)_q$ — базисной, т. е. ведущим элементом симплекс-таблицы является ξ_{pq}. Нас интересуют формулы пересчета величин γ_j для смежного базиса.

Рассмотрим вначале пересчет величины γ_q для ведущего столбца $(j = q)$. В соответствии с (3.27) имеем:

$$\overline{\gamma}_q = \overline{\delta}_q + \sum_{i \in \overline{R}} \overline{\xi}_{iq}^2.$$

Так как в смежном базисе $(x_B)_p$ и $(x_N)_q$ меняются местами, то $\overline{\eta}_p = \delta_q$, $\overline{\delta_q} = \eta_p$ и $\overline{R} \backslash \{p\} = R \backslash \{p\}$. Поэтому:

$$\overline{\gamma}_q = \overline{\delta}_q + \overline{\eta}_p \overline{\xi}_{pq}^2 + \sum_{i \in \overline{R} \backslash \{p\}} \overline{\xi}_{iq}^2 = \eta_p + \delta_q \overline{\xi}_{pq}^2 + \sum_{i \in R \backslash \{p\}} \overline{\xi}_{iq}^2.$$

Используя формулы пересчета (1.33)—(1.36), получим:

$$\overline{\gamma}_q = \eta_p + \delta_q \left(\frac{1}{\xi_{pq}}\right)^2 + \sum_{i \in R \backslash \{p\}} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2.$$

Заметим, что:

$$\sum_{i \in R} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = \eta_p \left(\frac{\xi_{pq}}{\xi_{pq}}\right)^2 + \sum_{i \in R \backslash \{p\}} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = \eta_p + \sum_{i \in R \backslash \{p\}} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2,$$

поэтому:

$$\overline{\gamma}_q = \delta_q \left(\frac{1}{\xi_{pq}}\right)^2 + \sum_{i \in R} \left(\frac{\xi_{iq}}{\xi_{pq}}\right)^2 = \frac{1}{\xi_{pq}^2} \left(\delta_q + \sum_{i \in R} \xi_{iq}^2\right),$$

откуда следует, что:

$$\overline{\gamma_q} = \frac{1}{\xi_{pq}^2} \gamma_q. \tag{3.28}$$

Рассмотрим теперь пересчет величин γ_j для остальных столбцов $(j \neq q)$. В соответствии с (3.27) имеем:

$$\overline{\gamma}_j = \overline{\delta}_j + \sum_{i \in \overline{R}} \overline{\xi}_{ij}^2.$$

Поскольку $j \neq q$, то $\overline{\delta}_j = \delta_j$. Кроме того, как уже было отмечено выше, $\overline{\eta}_p = \delta_q$ и $\overline{R}\backslash\{p\} = R\backslash\{p\}$. Поэтому:

$$\overline{\gamma}_j = \overline{\delta}_j + \overline{\eta}_p \overline{\xi}_{pj}^2 + \sum_{i \in \overline{R}\backslash\{p\}} \overline{\xi}_{ij}^2 = \delta_j + \delta_q \overline{\xi}_{pj}^2 + \sum_{i \in R\backslash\{p\}} \overline{\xi}_{ij}^2.$$

Используя формулы пересчета (1.33)—(1.36), получим:

$$\overline{\gamma}_j = \delta_j + \delta_q \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_{i \in R\backslash\{p\}} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2.$$

Заметим, что:

$$\sum_{i \in R} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 = \eta_p \left(\xi_{pj} - \frac{\xi_{pq}\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_{i \in R\backslash\{p\}} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2$$
$$= \sum_{i \in R\backslash\{p\}} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2,$$

поэтому:

$$\overline{\gamma}_j = \delta_j + \delta_q \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_{i \in R} \left(\xi_{ij} - \frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 =$$
$$= \delta_j + \delta_q \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 + \sum_{i \in R} \xi_{ij}^2 + \sum_{i \in R} \left(\frac{\xi_{iq}\xi_{pj}}{\xi_{pq}}\right)^2 - 2\sum_{i \in R} \frac{\xi_{ij}\xi_{iq}\xi_{pj}}{\xi_p q} =$$
$$= \left(\delta_j + \sum_{i \in R} \xi_{ij}^2\right) + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \left(\delta_q + \sum_{i \in R} \xi_{iq}^2\right) - 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) \sum_{i \in R} \xi_{ij}\xi_{iq},$$

откуда следует, что:

$$\overline{\gamma}_j = \gamma_j + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \gamma_q - 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) \sum_{i \in R} \xi_{ij}\xi_{iq}.$$

Сумма $\sum_{i \in R} \xi_{ij}\xi_{iq} = \sum_{i=1}^{m} \xi_{ij}\eta_i\xi_{iq}$ представляет собой скалярное произведение j-го столбца симплекс-таблицы $\Xi_j = (\xi_{ij}) = -B^{-1}N_j$ и вектора $H\Xi_q$, где $\Xi_q = (\xi_{iq}) = -B^{-1}N_q$ — q-й (ведущий) столбец симплекс-таблицы, $H = \mathrm{diag}(\eta_i)$ — диагональная матрица, составленная из величин η_i (3.17). Поэтому:

$$\sum_{i \in R} \xi_{ij}\xi_{iq} = \Xi_j^T (H\Xi_q) = -N_j^T B^{-T} H \Xi_q.$$

Таким образом, окончательно получаем:

$$\overline{\gamma}_j = \gamma_j + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \gamma_q + 2\left(\frac{\xi_{pj}}{\xi_{pq}}\right) N_j^T B^{-T} H \Xi_q. \tag{3.29}$$

Можно видеть, что формулы пересчета (3.28) и (3.29) совпадают с аналогичными формулами (3.11) и (3.12) для обычного метода наиболее крутого ребра. Единственное отличие состоит в том, что в формуле (3.12) обратное преобразование применяется непосредственно к ведущему столбцу симплекс-таблицы Ξ_q, а в формуле (3.29) — к вектору $H\Xi_q$, который получается заменой некоторых элементов ведущего столбца нулями. Таким образом, оба метода имеют одинаковую вычислительную трудоемкость.

Как и в случае обычного метода наиболее крутого ребра, многократный пересчет величин γ_j по формулам (3.28) и (3.29) может привести к отклонению этох величин от их точных значений вследствие ошибок округления. Поэтому, чтобы уменьшить влияние таких ошибок, можно применять ту же технику, что и ранее. Так, в формуле (3.28) рекомендуется использовать не текущее значение γ_q с предыдущей итерации, а более точное значение этой величины, вычисленное непосредственно по формуле (3.27):

$$\gamma_q = \delta_q + \sum_{i \in R} \xi_{iq}^2. \tag{3.30}$$

Кроме того, можно заметить, что:

$$\begin{aligned} \overline{\gamma}_j = \overline{\delta}_j + \sum_{i \in \overline{R}} \overline{\xi}_{ij}^2 = \delta_j + \delta_q \overline{\xi}_{pj}^2 + \sum_{i \in R \setminus \{p\}} \overline{\xi}_{ij}^2 \geq \\ \geq \delta_j + \delta_q \overline{\xi}_{pj}^2 = \delta_j + \delta_q \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2. \end{aligned} \tag{3.31}$$

Таким образом, если вследствие ошибок округления значение $\overline{\gamma}_j$, вычисленное по формуле (3.29), оказывается меньше, чем правая часть неравенства (3.31), то в качестве более точного значения $\overline{\gamma}_j$ имеет смысл использовать величину в правой части указанного неравенства.

Допустим, что p-я (ведущая) строка $\xi_p = (\xi_{pj})$ и q-й (ведущий) столбец $\Xi_q = (\xi_{iq})$ текущей симплекс-таблицы уже вычислены. Тогда практическая схема пересчета величин γ_j для смежного базиса в соответствии с проекционным методом наиболее крутого ребра может быть следующей.

1. Вычислить более точное значение γ_q для текущего базиса:

$$\gamma_q = \delta_q + \sum_{i \in R} \xi_{iq}^2.$$

2. Вычислить вспомогательный вектор, используя операцию BTRAN:

$$u = B^{-T}(H\Xi_q),$$

где $H = \mathrm{diag}(\eta_i)$ — диагональная матрица, составленная из величин η_i.

3. Выполнить пп. 4—6 для $j = 1, \ldots, q-1, q+1, \ldots, n-m$.

4. Вычислить вспомогательную величину:

$$r_j = \xi_{pj}/\xi_{pq}.$$

5. Вычислить вспомогательное скалярное произведение:

$$s_j = N_j^T u,$$

где N_j — j-й столбец матрицы N, т. е. столбец матрицы коэффициентов ограничений A, соответствующий небазисной переменной $(x_N)_j$.

6. Вычислить величину $\overline{\gamma}_j$ для смежного базиса:

$$\overline{\gamma}_j = \max(\gamma_j + r_j^2\gamma_q + 2r_j s_j,\ \delta_i + \delta_q r_j^2).$$

7. Вычислить величину $\overline{\gamma}_q$ для смежного базиса:

$$\overline{\gamma}_q = \gamma_q/\xi_{pq}^2.$$ ■

Последнее замечание касается использования правила (3.24). Чтобы не извлекать квадратный корень для вычислении нормы $\|\Delta_V^j\| = \gamma_j^{1/2}$ при выборе небазисной переменной $(x_N)_q$, можно использовать эквивалентное правило, которое совпадает с правилом (3.15) из подразд. 3.2:

$$\frac{d_q^2}{\gamma_q} = \max_{j \in J} \frac{d_j^2}{\gamma_j}. \tag{3.32}$$

Проекционный метод наиболее крутого ребра имеет те же положительные свойства, что и обычный метод наиболее крутого ребра в пространстве всех переменных. Однако возможность *переопределения* эталонного подпространства на любой итерации симплекс-метода позволяет избавиться от основного недостатка, присущего обычному методу наиболее крутого ребра и связанного с необходимостью вычисления всех столбцов симплекс-таблицы.

Начальное определение (инициализация) и переопределение эталонного подпространства возможны на любой итерации симплекс-метода и состоят в том, что эталонным подпространством становится текущее подпространство небазисных переменных. В соответствии с формулами (3.17) и (3.18) это означает, что $\eta_i = 0$ для всех $i = 1, \ldots, m$ (а значит, $R = \varnothing$) и $\delta_j = 1$ для всех $j = 1, \ldots, n-m$. Таким образом, как это следует из (3.27), сразу после инициализации или переопределения эталонного подпространства имеет место $\gamma_j = 1$ для всех $j = 1, \ldots, n-m$.

Поскольку проекционный метод наиболее крутого ребра основан на измерении углов в подпространстве, а не во всем пространстве переменных, то время от времени (например, через каждые 500 или 1000

итераций симплекс-метода) рекомендуется выполнять переопределение эталонного подпространства.

В заключение отметим, что проекционный метод наиболее крутого ребра по существу совпадает с методом оценивания Devex [8] (см. подразд. 3.4). Основное отличие состоит в том, что в данном методе весовые множители γ_j (3.27) пересчитываются точно, как это было предложено в статье [10] Х. Гринбергом (H. J. Greenberg) и Дж. Каланом (J. E. Kalan), а в методе Devex используется их достаточно грубая аппроксимация.

3.4 Метод оценивания Devex

Метод оценивания Devex — это метод выбора небазисной переменной, который был предложен П. Харрис (P. M. J. Harris) при разработке системы Devex [8]. Харрис первой обратила внимание на асимметрию правила Данцига и исходя из эмпирических соображений предложила измерять угол между выбираемым направлением и градиентом целевой функции не в текущем подпространстве небазисных переменных, которое изменяется на каждой итерации, а в фиксированном (эталонном) подпространстве.

Исторически метод оценивания Devex был первым методом выбора небазисной переменной, в котором используется идея эталонного (неизменного) подпространства. Этот метод по существу совпадает с проекционным методом наиболее крутого ребра (см. подразд. 3.3). Его единственное отличие состоит в том, что для пересчета масштабирующих величин γ_j (3.27)[24] Харрис использует приближенную формулу, которую можно получить из формулы (3.29), если отбросить последнее слагаемое:

$$\overline{\gamma}_j \approx \gamma_j + \left(\frac{\xi_{pj}}{\xi_{pq}}\right)^2 \gamma_q. \tag{3.33}$$

В данном случае требуется лишь одна операция обратного преобразования (BTRAN) на каждой итерации симплекс-метода для вычисления элементов ведущей строки симплекс-таблицы $\xi_p = (\xi_{pj})$,[25] поэтому использование метода оценивания Devex позволяет сэкономить одну операцию BTRAN на каждой итерации по сравнению с методами наиболее крутого ребра.

Допустим, что p-я (ведущая) строка $\xi_p = (\xi_{pj})$ и q-й (ведущий) столбец $\Xi_q = (\xi_{iq})$ текущей симплекс-таблицы уже вычислены. Тогда практическая схема пересчета величин γ_j для смежного базиса в соответствии с методом оценивания Devex может быть следующей. (Здесь считается, что эталонное подпространство V переменных определено посредством величин η_i (3.17) и δ_j (3.18) точно так же, как и в случае проекционного метода наиболее крутого ребра.)

1. Вычислить точное значение величины γ_q для текущего базиса:

$$\gamma_q = \delta_q + \sum_{i \in R} \xi_{iq}^2,$$

где $R = \{i : \eta_i = 1\} = \{i : (x_B)_i \in V\}$ — множество индексов базисных переменных, принадлежащих эталонному подпространству. (Заметим, что

[24] В статье [8] вместо величин γ_j используются сами нормы $T_j = \sqrt{\gamma_j} = \|\Delta x_V^j\|$, которые Харрис называет «tread» (англ. — ширина хода).

[25] Эта строка также используется для пересчета относительных оценок небазисных переменных (см. подразд. 2.3).

в соответствии с формулой (3.33) величины γ_j монотонно возрастают. Поэтому если вычисленное точное значение γ_q значительно отличается от его текущего приближенного значения, следует переопределить эталонное подпространство.)

2. Вычислить величины $\overline{\gamma_j},\ j = 1, \ldots, q-1, q+1, \ldots, n-m$ для смежного базиса:

$$\overline{\gamma_j} = \gamma_j + (\xi_{pj}/\xi_{pq})^2 \gamma_q.$$

3. Вычислить величину $\overline{\gamma_q}$ для смежного базиса:

$$\overline{\gamma_q} = \gamma_q / \xi_{pq}^2.$$ ■

Выбор небазисной переменной $(x_N)_q$ выполняется по правилу (3.32) аналогично тому, как это делается при использовании проекционного метода наиболее крутого ребра. Начальное определение (инициализация) и переопределение эталонного подпространства также выполняются аналогичным образом.

Литература

[1] G. B. Dantzig. Programming of Interdependent Activities, II, Mathematical Model, pp. 19-32, in: Activity Analysis of Production and Allocation, ed. T. C. Koopmans. New York: John Wiley & Sons, Inc., 1951. См. также: Econometrics 17, No. 3, 4 (July — Oct. 1949), pp. 200-211.

[2] G. B. Dantzig. Linear Programming and Extensions. Princeton, N. J.: Princeton University Press, 1963. [Имеется перевод: Дж. Данциг. Линейное программирование, его применения и обобщения.— М.: Прогресс, 1966.]

[3] G. B. Dantzig. Computational Algorithm of the Revised Simplex Method. RAND RM-1266, 1953.

[4] G. B. Dantzig, W. Orchard-Hays. Alternate Algorithm for the Revised Simplex Method. RAND RM-1268, 1953.

[5] G. B. Dantzig, A. Orden, P. Wolfe. The Generalized Simplex Method. RAND RM-1264, 1954.

[6] Mathematical Programming System/360 (360A-CO-14X). N. Y.: IBM Corp., 1966.

[7] R. E. Bixby. Progress in linear programming. ORSA J. on Computing, 6, 1994, pp. 15-22.

[8] P. M. J. Harris. Pivot Selection Methods of the Devex LP Code. Mathematical Programming 12, 1973, pp. 1-28.

[9] D. Goldfarb, J. K. Reid. A Practicable Steepest-Edge Simplex Algorithm. Mathematical Programming 12, 1977, pp. 361-371.

[10] H. J. Greenberg, J. E. Kalan. An exact update for Harris' TREAD. Mathematical Programming Study 4, 1975, pp. 26-29.