

УДК: 519.85

## Оценка числа итераций для сильно полиномиальных алгоритмов линейного программирования

А. Б. Свириденко

ФГБОУ ВПО «Кубанский государственный университет», филиал в г. Новороссийске,  
Россия, 353922, г. Новороссийск, ул. Героев Десантников, д. 87

E-mail: roshechka@gmail.com

*Получено 04.05.2022, после доработки — 17.10.2023.  
Принято к публикации 20.11.2023.*

Рассматривается прямой алгоритм решения задачи линейного программирования (ЛП), заданной в каноническом виде. Алгоритм состоит из двух последовательных этапов, на которых прямым методом решаются приведенные ниже задачи ЛП: невырожденная вспомогательная задача (на первом этапе) и некоторая задача, равносильная исходной (на втором). В основе построения вспомогательной задачи лежит мультипликативный вариант метода исключения Гаусса, в самой структуре которого заложены возможности: идентификации несовместности и линейной зависимости ограничений; идентификации переменных, оптимальные значения которых заведомо равны нулю; фактического исключения прямых переменных и сокращения размерности пространства, в котором определено решение исходной задачи. В процессе фактического исключения переменных алгоритм генерирует последовательность мультипликаторов, главные строки которых формируют матрицу ограничений вспомогательной задачи, причем возможность минимизация заполнения главных строк мультипликаторов заложена в самой структуре прямых методов. При этом отсутствует необходимость передачи информации (базис, план и оптимальное значение целевой функции) на второй этап алгоритма и применения одного из способов устранения зацикливания для гарантии конечной сходимости.

Представлены два варианта алгоритма решения вспомогательной задачи в сопряженной канонической форме. Первый основан на ее решении прямым алгоритмом в терминах симплекс-метода, а второй — на решении задачи, двойственной к ней, симплекс-методом. Показано, что оба варианта алгоритма для одинаковых исходных данных (входов) генерируют одинаковую последовательность точек: базисное решение и текущее двойственное решение вектора оценок строк. Отсюда сделан вывод, что прямой алгоритм — это алгоритм типа симплекс-метода. Также показано, что сравнение вычислительных схем приводит к выводу, что прямой алгоритм позволяет уменьшить по кубическому закону число арифметических операций, необходимых для решения вспомогательной задачи, по сравнению с симплекс-методом. Приводится оценка числа итераций.

Ключевые слова: линейное программирование, алгоритм симплекс-метода, прямой алгоритм, число итераций, сильно полиномиальный алгоритм

UDC: 519.85

## The iterations' number estimation for strongly polynomial linear programming algorithms

A. B. Sviridenko

FSEI of HPE "Kuban State University" branch in Novorossiysk,  
87 Geroev Desantnikov st., Novorossiysk, 353922, Russia

E-mail: roshechka@gmail.com

*Received 04.05.2022, after completion — 17.10.2023.*

*Accepted for publication 20.11.2023.*

A direct algorithm for solving a linear programming problem (LP), given in canonical form, is considered. The algorithm consists of two successive stages, in which the following LP problems are solved by a direct method: a non-degenerate auxiliary problem at the first stage and some problem equivalent to the original one at the second. The construction of the auxiliary problem is based on a multiplicative version of the Gaussian exclusion method, in the very structure of which there are possibilities: identification of incompatibility and linear dependence of constraints; identification of variables whose optimal values are obviously zero; the actual exclusion of direct variables and the reduction of the dimension of the space in which the solution of the original problem is determined. In the process of actual exclusion of variables, the algorithm generates a sequence of multipliers, the main rows of which form a matrix of constraints of the auxiliary problem, and the possibility of minimizing the filling of the main rows of multipliers is inherent in the very structure of direct methods. At the same time, there is no need to transfer information (basis, plan and optimal value of the objective function) to the second stage of the algorithm and apply one of the ways to eliminate looping to guarantee final convergence.

Two variants of the algorithm for solving the auxiliary problem in conjugate canonical form are presented. The first one is based on its solution by a direct algorithm in terms of the simplex method, and the second one is based on solving a problem dual to it by the simplex method. It is shown that both variants of the algorithm for the same initial data (inputs) generate the same sequence of points: the basic solution and the current dual solution of the vector of row estimates. Hence, it is concluded that the direct algorithm is an algorithm of the simplex method type. It is also shown that the comparison of numerical schemes leads to the conclusion that the direct algorithm allows to reduce, according to the cubic law, the number of arithmetic operations necessary to solve the auxiliary problem, compared with the simplex method. An estimate of the number of iterations is given.

Keywords: linear programming, simplex method algorithm, direct algorithm, number of iterations, strongly polynomial algorithm

Citation: *Computer Research and Modeling*, 2024, vol. 16, no. 2, pp. 249–285 (Russian).

## Введение

Рассматривается задача ЛП в канонической форме:

$$\min_{x \in R^n}(cx), \quad h + Hx = 0, \quad x \geq 0, \quad (1)$$

где  $H$  —  $(m \times n)$ -матрица, в которой  $m < n$ ,  $c^T \in R^n$ ,  $h \in R^m$ ,  $x$  — вектор прямых переменных, элементы векторов  $c^T$ ,  $h$  и матрицы  $H$  — произвольные вещественные числа.

Конечный алгоритм решения задачи (1), основанный на симплекс-методе с обратной матрицей (модифицированный симплекс-метод), состоит из двух последовательных этапов, на которых симплекс-методом решаются задачи ЛП: вспомогательная задача (на первом этапе) и некоторая задача, равносильная исходной (на втором) [Агафонова, Даугавет, 2017]. Алгоритм не требует ни предварительных расчетов, ни наложения каких-либо дополнительных условий на  $c$ ,  $h$ ,  $H$ , в том числе условия невырожденности задачи, обеспечивающего конечную сходимость алгоритма. Конечную сходимость гарантирует применение правила Блэнда для предотвращения заикливания или любой другой способ устранения заикливания. В основе алгоритма лежит поиск решения среди вершин многогранника допустимых решений. Принцип поиска состоит в следующем. Сначала находится одна из вершин, а затем ищется ближайшая к ней вершина, в которой значение целевой функции убывает. Как только переход от одной вершины к другой с меньшим значением целевой функции становится невозможным, делается вывод, что решение найдено. Основным недостатком алгоритма — его сложность. В основном это необходимость передачи информации (базис, план и оптимальное значение целевой функции [Агафонова, Даугавет, 2017]) на следующий этап алгоритма и применения одного из способов устранения заикливания для гарантии конечной сходимости.

В данной работе, являющейся продолжением исследований [Свириденко, 2015; Свириденко, 2016; Свириденко, 2017; Свириденко, 2019], в основу устранения недостатков симплекс-метода с обратной матрицей положен отказ от поиска решения среди вершин многогранника допустимых решений и переход к прямому поиску. Для упрощения будем считать, что  $c \geq 0$ , иначе в описании будут участвовать следующие вспомогательные величины:

$c_0$  ( $-c_0 \leq cx^*$ ) — сколь угодно большая скалярная величина,

$x_0$  ( $x_0 \geq 0$ ) — дополнительная переменная,

$c_0 - x_0 + cx = 0$  — уравнение связи дополнительной переменной,

$\min_{x \in R^n}(x_0)$ ,  $c_0 - x_0 + cx = 0$ ,  $h + Hx = 0$ ,  $x_0 \geq 0$ ,  $x \geq 0$ , — задача, эквивалентная (1).

Технически это делается так, как показано в описании вычислительной схемы первого этапа прямого алгоритма в терминах симплекс-метода в подпараграфе 3.2 и на примерах решения задач с гарантированным поведением симплекс-метода в § 4. Также будем считать, что задача (1), быть может вырожденная, имеет единственное решение, либо любая допустимая точка является оптимальной, причем последнее имеет место в том и только в том случае, если  $c$  есть линейная комбинация строк матрицы  $H$ , т. е. существует  $m$ -мерная вектор-строка  $w$  такая, что  $c = wH$ . Отсюда вытекает, что целевая функция задачи (1) является константой ( $cx = wHx = -wh$ ), а константу оптимизировать невозможно. Строго говоря, такая задача не является задачей ЛП, это классическая задача на условный экстремум из курса математического анализа [Беников, 2005].

## 1. Обоснование алгоритма

Обозначим:

$k$  — номер итерации,

$h_{k \bullet} = (h_{k1} \quad h_{k2} \quad \dots \quad h_{kn})$  —  $k$ -я строка матрицы  $H$ ,

$h_k^k + (h_{kk}^k \ h_{kk+1}^k \ \dots \ h_{kn}^k)(x_k \ x_{k+1} \ \dots \ x_n)^T = 0$  — уравнение связи  $k$ -й строки,  
 $x^k = (x_1^k \ x_2^k \ \dots \ x_k^k \ 0 \ \dots \ 0)^T$  —  $n$ -мерный вектор, решение системы линейных уравнений:

$$h_j + h_{j\bullet} x = 0, \quad j = 1, \dots, k,$$

$e_{1_k\bullet}^k = (e_{1_k k+1}^k \ e_{1_k k+2}^k \ \dots \ e_{1_k n}^k)$  — главная строка мультипликатора:

$$E_{1_k}^k = \begin{pmatrix} e_{1_k k+1}^k & e_{1_k k+2}^k & \dots & e_{1_k n}^k \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}.$$

Здесь  $E_{1_k}^k$  —  $(n-k) \times (n-k-1)$ -матрица, первая строка которой произвольная, а остальные — строки единичной  $(n-k-1) \times (n-k-1)$ -матрицы.  $E_{r_k}^k$  ( $r_k$  — номер ведущей строки на итерации  $k$ ) совпадает с мультипликатором в алгоритме симплекс-метода с точностью до знака транспонирования [Хакимова, Зеленков, Рзун, 2010; Свириденко, 2015; Свириденко, 2017], за исключением удаления ведущего столбца, все элементы которого равны нулю. Представленная форма мультипликатора  $E_{1_k}^k$  является артефактом и вызвана предположением  $r = k$ , которое сделано ниже на шаге 3 первого этапа алгоритма.

При введенных обозначениях принцип поиска решения двухэтапным алгоритмом состоит в следующем. Многогранник допустимых решений задается не изначально, как в симплекс-методе, а формируется построением на каждой итерации  $k = 1, \dots, m$  задачи:

$$\begin{aligned} & \min_{x \in R^n} (c_{k+1}^k x_{k+1} + c_{k+2}^k x_{k+2} + \dots + c_n^k x_n), \\ x_i &= x_i^k + e_{1_k k+1}^i x_{k+1} + e_{1_k k+2}^i x_{k+2} + \dots + e_{1_k n}^i x_n, \quad i = 1, \dots, k, \\ x_i &\geq 0, \quad i = 1, \dots, n, \end{aligned} \quad (2)$$

в канонической форме, эквивалентной задаче:

$$\min_{x \in R^n} (cx), \quad h_j + h_{j\bullet} x = 0, \quad j = 1, \dots, k, \quad x \geq 0.$$

В основе построения (2) лежит мультипликативный вариант метода исключения Гаусса [Свириденко, 2016], в самой структуре которого заложены возможности: идентификация несовместности и линейной зависимости ограничений; идентификация переменных, оптимальные значения которых заведомо равны нулю; фактическое исключение прямых переменных и сокращение размерности пространства, в котором определено решение задачи (1). Перечисленные возможности определяют построение (2) на первом этапе алгоритма при обязательном выполнении условий:

$$x_i = x_i^k \geq 0, \quad i = 1, \dots, k.$$

Необязательность выполнения всех ограничений для текущего решения обеспечивает большую гибкость алгоритма в случае «вылетов» из допустимой области и использования найденного решения в качестве начального для той же задачи с измененными условиями.

Начальным является многогранник, задаваемый условиями на неотрицательность переменных, на котором минимальное значение целевой функции равно нулю. Затем в его систему условий добавляется первое ограничение-равенство и вычисляется оптимум на сформированном многограннике, увеличивая (в силу сжатия области допустимых решений) значение целевой

функции на некоторую неотрицательную константу. Вычисления продолжают до тех пор, пока на итерации  $t$  в систему условий не будет добавлено последнее ограничение. Первый этап алгоритма описан, технически на каждой итерации это делается так (детальное описание всех операций, выполняемых по ходу построения (2), приведено в подпараграфе 2.1).

### 1.1. Первый этап алгоритма

**Шаг 1.** Построить уравнения связи:

$$h_k^k + h_{kk}^k x_k + h_{kk+1}^k x_{k+1} + \dots + h_{kn}^k x_n = 0$$

$k$ -й строки системы ограничений:

$$h_k + h_k \bullet x = 0.$$

Если  $k = 1$ , то положить

$$h_k^k = h_k, \quad h_{ki}^k = h_{ki}, \quad i = k, \dots, n.$$

Если  $k \geq 2$ , то вычислить по формулам

$$h_k^k = -h_k + \sum_{i=1}^{k-1} h_{ki} x_i, \quad (h_{kk}^k \quad h_{kk+1}^k \quad \dots \quad h_{kn}^k) = h_k \bullet \prod_{i=1}^{k-1} E_{1_{k-1}}^i.$$

ЗАМЕЧАНИЕ. Формулы шага 1 — это результат подстановки:

$$x_i = x_i^k + e_{1_k k+1}^i x_{k+1} + e_{1_k k+2}^i x_{k+2} + \dots + e_{1_k n}^i x_n, \quad i = 1, \dots, k-1,$$

в  $k$ -ю строку системы ограничений.

Не нарушая общности, для упрощения будем считать

$$h_k^k \leq 0,$$

иначе достаточно положить

$$h_k^k = -h_k^k, \quad h_{ki}^k = -h_{ki}^k, \quad i = k, \dots, n.$$

**Шаг 2.** Анализ уравнения связи (идентификация несовместности и линейной зависимости ограничений; идентификация переменных, оптимальные значения которых заведомо равны нулю).

**Шаг 3.** Вычислить номер  $r$  ведущего столбца по правилу:

$$\Theta_r = \min_i \frac{c_i^{k-1}}{h_{ki}^k}, \quad h_{ki}^k > 0, \quad i = k, \dots, n.$$

Не нарушая общности, будем считать

$$r = k.$$

Переписать уравнение связи:

$$x_r = x_k = x_k^k + e_{1_k}^k \bullet (x_{k+1} \quad x_{k+2} \quad \dots \quad x_n)^T.$$

Если  $k = 1$ , то положить

$$c^k = (c_{k+1}^k \quad c_{k+2}^k \quad \dots \quad c_n^k) = c E_{1_k}^k, \\ x^k = (x_1^k \quad 0 \quad \dots \quad 0)^T.$$

Если  $k \geq 2$ , то вычислить по формулам

$$\begin{aligned} c^k &= (c_{k+1}^k \quad c_{k+2}^k \quad \cdots \quad c_n^k) = c^{k-1} E_{1_k}^k, \\ e_{1_k}^i \bullet &= e_{1_{k-1}}^i \bullet E_{1_k}^k, \quad x_i^k = x_i^i + x_k^k e_{1_{k-1}k}^i, \quad i = 1, \dots, k-1, \\ x^k &= (x_1^k \quad x_2^k \quad \cdots \quad x_k^k \quad 0 \quad \cdots \quad 0)^T. \end{aligned}$$

ЗАМЕЧАНИЕ. Формулы шага 3 — это результат подстановки  $x_k$  в условия задачи (2).

Правило выбора ведущего столбца гарантирует положительность  $c^k$  на каждой итерации  $k = 1, \dots, m$ , поэтому, если  $x^m \geq 0$ , то алгоритм становится одноэтапным (см. пример 1 из § 4 на заикливание симплекс-метода); иначе переходим к описанию второго этапа.

## 1.2. Второй этап алгоритма

Для упрощения описания обозначим через  $\nu$  число итераций и для  $\nu = 0$  положим

$$\begin{aligned} x &= x_\nu = (x_{\nu_1} \quad x_{\nu_2} \quad \cdots \quad x_{\nu_n})^T, \\ x^m &= x_\nu^m = (x_{\nu_1}^\nu \quad x_{\nu_2}^\nu \quad \cdots \quad x_{\nu_m}^\nu \quad 0 \quad \cdots \quad 0)^T, \\ c^m &= c_\nu^m = (c_{\nu_{m+1}}^\nu \quad c_{\nu_{m+2}}^\nu \quad \cdots \quad c_{\nu_n}^\nu), \\ e_{1_m}^i \bullet &= e_{\nu_i}^m \bullet = (e_{\nu_{m+1}}^{\nu_i} \quad e_{\nu_{m+2}}^{\nu_i} \quad \cdots \quad e_{\nu_n}^{\nu_i}), \quad i = 1, \dots, m. \end{aligned}$$

При введенных обозначениях перепишем (2) в виде

$$\begin{aligned} \min_{x_\nu \in R^n} & (c_{\nu_{m+1}}^\nu x_{\nu_{m+1}} + c_{\nu_{m+2}}^\nu x_{\nu_{m+2}} + \cdots + c_{\nu_n}^\nu x_{\nu_n}), \\ x_{\nu_i} &= x_{\nu_i}^\nu + e_{\nu_{m+1}}^{\nu_i} x_{\nu_{m+1}} + e_{\nu_{m+2}}^{\nu_i} x_{\nu_{m+2}} + \cdots + e_{\nu_n}^{\nu_i} x_{\nu_n}, \quad i = 1, \dots, m, \\ x_{\nu_i} &\geq 0, \quad i = 1, \dots, n. \end{aligned} \quad (3)$$

С математической точки зрения (3) эквивалентна задаче в сопряженной канонической форме:

$$\begin{aligned} \min_{x_\nu \in R^{n-m}} & (c_{\nu_{m+1}}^\nu x_{\nu_{m+1}} + c_{\nu_{m+2}}^\nu x_{\nu_{m+2}} + \cdots + c_{\nu_n}^\nu x_{\nu_n}), \\ x_{\nu_i}^\nu + e_{\nu_{m+1}}^{\nu_i} x_{\nu_{m+1}} + e_{\nu_{m+2}}^{\nu_i} x_{\nu_{m+2}} + \cdots + e_{\nu_n}^{\nu_i} x_{\nu_n} &\geq 0, \quad i = 1, \dots, m, \\ x_{\nu_i} &\geq 0, \quad i = m+1, \dots, n. \end{aligned} \quad (4)$$

ЗАМЕЧАНИЕ.  $x_{\nu_i}$  в (3) можно рассматривать как дополнительные, их значения равны  $x_{\nu_i}^\nu$ . В подпараграфе 2.3 показано, что условия (3) переопределены и что, по существу, (3) и (4) — одна и та же задача. Более того, известно допустимое решение задачи, двойственной к (4), — нулевой  $m$ -мерный вектор, алгоритм решения которой рассматривается в третьем разделе.

На каждой итерации второго этапа в расчет вводится ограничение по формуле

$$x_{\nu_i} = x_{\nu_i}^\nu + e_{\nu_{m+1}}^{\nu_i} x_{\nu_{m+1}} + e_{\nu_{m+2}}^{\nu_i} x_{\nu_{m+2}} + \cdots + e_{\nu_n}^{\nu_i} x_{\nu_n}, \quad x_{\nu_i}^\nu < 0,$$

увеличивая (в силу сжатия области допустимых решений) значение целевой функции на некоторую положительную константу. Вычисления продолжаются до тех пор, пока не будет найдено минимальное значение целевой функции на многограннике окончательной конфигурации. Технически на каждой итерации это делается так (детальное описание всех операций, выполняемых по ходу решения (3), приведено в подпараграфе 2.2).

**Шаг 1.** Вычислить номер  $v_q$  ведущего уравнения связи по формуле

$$x_{v_q} = x_{v_q}^v + e_{v_q, m+1}^v x_{v_{m+1}} + e_{v_q, m+2}^v x_{v_{m+2}} + \dots + e_{v_q, n}^v x_{v_n}, \quad x_{v_q}^v < 0.$$

Вычислить номер  $v_r$  ведущего столбца по правилу

$$\Theta_{v_r} = \min_i \frac{c_{v_i}^v}{e_{v_q i}^v}, \quad e_{v_q i}^v > 0, \quad i = m + 1, \dots, n.$$

Положить  $v = v + 1$ .

**Шаг 2.** Вычислить

$$x_{v_r}^v = -\frac{x_{v-1, q}^{v-1}}{e_{v-1, q r}^{v-1}}, \quad x_{v_i}^v = x_{v-1, i}^{v-1} + e_{v-1, i r}^{v-1} x_{v_r}^v, \quad i \neq r.$$

Положить

$$x_v^v = (x_{v_1}^v \quad x_{v_2}^v \quad \dots \quad x_{v_m}^v \quad 0 \quad \dots \quad 0)^T.$$

Если  $x_{v_r}^v \geq 0$ , то остановиться.

**Шаг 3.** Переписать уравнение связи:

$$x_{v_r} = x_{v_r}^v + e_{v_r, m+1}^v x_{v_{m+1}} + e_{v_r, m+2}^v x_{v_{m+2}} + \dots + e_{v_r, n}^v x_{v_n}.$$

Вычислить по формулам

$$e_{v_i \bullet}^v = e_{v-1, i \bullet}^{v-1} E_{v_r}^v, \quad i \neq q,$$

$$c_{v_i}^v = (c_{v_{m+1}}^v \quad c_{v_{m+2}}^v \quad \dots \quad c_{v_n}^v) = c_{v-1}^{v-1} E_{v_r}^v$$

и перейти к шагу 1.

В случае неединственности решения прямой алгоритм, как и симплекс-метод, дает решение, которое принадлежит вершине многогранного множества. Не нарушая общности, будем считать

$$x_{v_i}^v \geq 0, \quad c_{v_i}^v = 0 \quad (i = m + 1, \dots, m_0), \quad c_{v_i}^v > 0 \quad (i = m_0 + 1, \dots, n).$$

Это означает, что оптимальные значения переменных

$$x_{v_i} \quad (i = m_0 + 1, \dots, n)$$

заведомо равны нулю, поэтому их можно исключить из рассмотрения. Следовательно, любое решение системы

$$x_{v_i} = x_{v_i}^v + e_{v_i, m+1}^v x_{v_{m+1}} + e_{v_i, m+2}^v x_{v_{m+2}} + \dots + e_{v_i, n}^v x_{v_n} \quad (i = 1, \dots, m), \quad x_{v_i} \geq 0 \quad (i = 1, \dots, m_0)$$

является оптимальным. Таким образом, неединственность решения означает неопределенность и возможность выбора наилучшего по тому или иному дополнительному критерию, исходя из неформализованных (эвристических) представлений об объекте. Известно, что метод внутренних точек и метод квадратичной штрафной функции приводят к различным решениям в случае их неединственности. Методы внутренней точки сходятся к решению, в котором выполнено условие строгой дополняющей нежесткости. Метод внешней квадратичной штрафной функции дает возможность найти решение, обладающее минимальной евклидовой нормой. В работе показано, что прямой алгоритм дает возможность найти все оптимальные вершины многогранного множества и, следовательно, приводит к точному нормальному решению, и к решению, в котором выполнено условие дополняющей нежесткости (см. пример 4 из § 4 и заключение).

### 1.3. Замечания к разделу

Прямой метод представляет собой последовательность шагов, на каждом из которых получают нули в нужных позициях очередного обрабатываемого столбца матрицы условий прямой задачи. При этом сохраняются нули, полученные ранее в предыдущих столбцах. Это определение полностью соответствует построению (3) на первом этапе и пересчету главных строк мультипликаторов — на втором. Правило выбора ведущего столбца гарантирует неотрицательность  $c_v^y$  на каждой итерации  $v$ , а единственность решения (1), равносильной (3), — их положительность. Исключение «лишних» ограничений и переменных, оптимальные значения которых заведомо равны нулю, гарантирует невырожденность (3), а строгое возрастание целевой функции от итерации к итерации — конечность числа итераций. Таким образом, отсутствует необходимость передачи информации на второй этап и применения одного из способов устранения зацикливания для гарантии конечной сходимости.

Задачи (2), генерируемые при формировании многогранника допустимых решений, обуславливают варианты построения прямого алгоритма. Например, для одноэтапного варианта достаточно решения (2) на каждой итерации, если  $x^k \not\geq 0$ . Иными словами, путем объединения этапов на каждой итерации  $k$ . Одноэтапные алгоритмы в данной работе не рассматриваются, так как исключают возможность построения двухэтапного варианта сильно полиномиального симплекс-метода с обратной матрицей.

На каждой итерации каждого этапа алгоритм генерирует последовательность точек, выходящих за пределы допустимой области в направлении возрастания целевой функции, до тех пор, пока не будет получено допустимое решение, причем гиперплоскость целевой функции, проходящая через эти точки, не пересекает допустимую область. Поэтому в соответствии с терминологией Фиакко и Мак-Кормика его можно назвать одним из вариантов метода внешних точек [Фиакко, Мак-Кормик, 1972].

В подпараграфе 2.1 описана вычислительная схема построения (2). В процессе построения алгоритм генерирует последовательность мультипликаторов, главные строки которых формируют матрицу ограничений (2). Минимизация их заполнения — основная задача при построении алгоритма. Возможность такой минимизации без потери точности результатов заложена в самой структуре прямых методов, поэтому вычислительную схему первого этапа можно рассматривать как реализацию алгоритма сокращения размерности задачи. В подпараграфе 2.2 приведена вычислительная схема решения (3) за сильно полиномиальное время, т. е. число арифметических операций (однородная сложность) ограничено полиномом от размерности задачи [Кузюрин, Фомин, 2007]. Для доказательства достаточно показать, что итераций не превосходит  $\frac{n}{2}$ . В подпараграфе 2.3 представлены формы записи (3): каноническая и сопряженная с канонической. В подпараграфе 2.4 даны геометрическая интерпретация (3), а также геометрическое представление итераций симплекс-метода и прямого алгоритма. В § 3 рассматриваются взаимосвязь прямого алгоритма с модифицированным симплекс-методом и построение верхней оценки числа итераций. В подпараграфе 3.1 дана вычислительная схема второго этапа модифицированного симплекс-метода, связанная с выделением базисной системы ограничений независимо от метода ее построения. В подпараграфах 3.2, 3.3 полученные результаты применяются для построения вычислительных схем этапов прямого алгоритма в терминах симплекс-метода. В подпараграфе 3.4 приведено сравнение прямого алгоритма решения (4) с симплекс-методом решения задачи, двойственной к (4). Показано, что алгоритмы для одинаковых исходных данных (входов) генерируют одинаковую последовательность точек: базисное решение и текущее двойственное решение вектора оценок строк. Это означает, что прямой алгоритм — это вариант симплекс-метода. Там же показано, что сравнение вычислительных схем приводит к выводу, что прямой алгоритм позволяет уменьшить по кубическому закону число арифметических операций, необходимых для решения задачи, двойственной к (4), по сравнению с симплекс-методом. В подпараграфе 3.5



полученные результаты применяются для построения верхней оценки итераций симплекс-метода решения задачи, двойственной к (4). Показано, что число итераций не превосходит  $\frac{m}{2}$ . Это означает, что симплекс-метод, допускающий решение задач вида (1) за сильно полиномиальное время, можно построить из двух последовательных этапов, на которых решаются приведенные ниже задачи ЛП: построение (3) прямым алгоритмом на первом этапе и решение задачи, двойственной к (3) в сопряженной канонической форме, симплекс-методом — на втором.

## 2. Предварительные сведения, формы записи и геометрическая интерпретация

Ниже приводится детальное описание всех операций, выполняемых по ходу построения (3). Если в процессе вычислений элементы становятся меньше по абсолютной величине, так называемого критического значения  $\varepsilon_0$ , то их предлагается приравнять нулю.

### 2.1. Первый этап алгоритма

Обозначим:

$\mathbf{x}^*$  —  $n$ -мерный вектор, решение задачи (1).

**Шаг 0** (инициализация). Положить

$$k = 1, \\ c^{k-1} = c = (c_1 \quad c_2 \quad \dots \quad c_n).$$

Вычислить номер  $q$  ведущей строки:

$$\theta_q = \max_j |h_j|, \quad j = k, \dots, m.$$

Поменять местами элементы  $q$ -й и  $k$ -й строк матрицы  $H$  и вектора  $h$ . Перенумеровать и запомнить порядок строк.

Вычислить элементы  $h_k^k, h_{ki}^k$  уравнения связи  $k$ -й строки:

$$h_k^k + (h_{kk}^k \quad h_{kk+1}^k \quad \dots \quad h_{kn}^k)(x_k \quad x_{k+1} \quad \dots \quad x_n)^T = 0, \\ h_k^k = h_k, \quad (h_{kk}^k \quad h_{kk+1}^k \quad \dots \quad h_{kn}^k) = (h_{kk} \quad h_{kk+1} \quad \dots \quad h_{kn}).$$

Здесь и далее будем считать  $h_k^k \leq 0$ , иначе достаточно положить

$$h_k^k = -h_k^k, \quad h_{ki}^k = -h_{ki}^k, \quad i = k, \dots, n.$$

Вычислить номер  $r$  ведущего столбца:

$$\Theta_r = \min_i \frac{c_i^{k-1}}{h_{ki}^k}, \quad h_{ki}^k > 0, \quad i = k, \dots, n.$$

Поменять местами элементы  $r$ -го и  $k$ -го столбцов матрицы  $H$  и вектор-строки  $c^0$ . Перенумеровать и запомнить порядок неизвестных.

Переписать уравнение связи:

$$x_k = x_k^k + e_{1_k}^k \cdot (x_{k+1} \quad x_{k+2} \quad \dots \quad x_n)^T, \\ x_k^k = -\frac{h_k^k}{h_{kk}^k}, \quad e_{1_k}^k = -\frac{(h_{kk+1}^k \quad h_{kk+2}^k \quad \dots \quad h_{kn}^k)}{h_{kk}^k}.$$

Вычислить

$$c^k = (c_{k+1}^k \quad c_{k+2}^k \quad \dots \quad c_n^k) = c^{k-1} E_{1_k}^k.$$

Положить

$$x^k = (x_1^k \quad 0 \quad \dots \quad 0)^T.$$

Перейти к шагу 6.

**Шаг 1** (расчет  $h_k^k, h_{ki}^k$ ). Вычислить номер  $q$  ведущей строки по формуле

$$\theta_q = \max_j |-h_j + h_{j\bullet} x^{k-1}|, \quad j = k, \dots, m.$$

Поменять местами элементы  $q$ -й и  $k$ -й строк матрицы  $H$  и вектора  $h$ . Перенумеровать и запомнить порядок строк.

Вычислить:

$$h_k^k = -h_k + \sum_{i=1}^{k-1} h_{ki} x_i^i, \quad (h_{kk}^k \quad h_{kk+1}^k \quad \dots \quad h_{kn}^k) = h_{k\bullet} \cdot \prod_{i=1}^{k-1} E_{1_{k-1}}^i = h_{k\bullet} \cdot \begin{pmatrix} e_{1_{k-1}k}^1 & \dots & e_{1_{k-1}n}^1 \\ \vdots & \ddots & \vdots \\ e_{1_{k-1}k}^{k-1} & \dots & e_{1_{k-1}n}^{k-1} \\ 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}.$$

ЗАМЕЧАНИЕ. Для ручного счета стратегия выбора ведущей строки может быть основана на упрощении вычислений.

**Шаг 2** (анализ  $h_k^k, h_{ki}^k$ ). Без ограничения общности при анализе элементов уравнения связи можно исключить из дальнейшего рассмотрения идентификацию следующих ситуаций:

- $h_k^k < 0, h_{ki}^k \leq 0, i = k, \dots, n$ , — несовместности ограничений задачи (1);
- $h_k^k = 0, h_{ki}^k = 0, i = k, \dots, n$ , — исключения  $k$ -й строки из числа ограничений задачи (1) и, как следствие, уменьшения числа мультипликаторов;
- $h_k^k = 0, h_{ki}^k \leq 0$  (или  $h_{ki}^k \geq 0$ ),  $i = k, \dots, n$ , — уменьшения размерности задачи (1) и, как следствие, уменьшения числа ненулевых элементов главных строк мультипликаторов, полученных на предыдущих шагах.

**Шаг 3** (расчет  $x_k^k, e_{1_k}^k$ ). Вычислить номер  $r$  ведущего столбца:

$$\Theta_r = \min_i \frac{c_i^{k-1}}{h_{ki}^k}, \quad h_{ki}^k > 0, \quad i = k, \dots, n.$$

Поменять местами элементы  $r$ -го и  $k$ -го столбцов матрицы  $H$  и вектор-строки  $c^0$ , перенумеровать и запомнить порядок неизвестных.

Переписать уравнение связи:

$$x_k = x_k^k + e_{1_k}^k \cdot (x_{k+1} \quad x_{k+2} \quad \dots \quad x_n)^T,$$

$$x_k^k = -\frac{h_k^k}{h_{kk}^k}, \quad e_{1_k}^k = -\frac{(h_{kk+1}^k \quad h_{kk+2}^k \quad \dots \quad h_{kn}^k)}{h_{kk}^k}.$$

**Шаг 4** (расчет  $c^k, x^k$ ). Вычислить

$$c^k = (c_{k+1}^k \quad c_{k+2}^k \quad \dots \quad c_n^k) = c^{k-1} E_{1_k}^k, \quad x_i^k = x_i^{i-1} + x_k^k e_{1_{k-1}k}^i, \quad i = 1, \dots, k-1.$$

Положить

$$x^k = (x_1^k \quad x_2^k \quad \dots \quad x_k^k \quad 0 \quad \dots \quad 0)^T.$$

**Шаг 5** (пересчет  $e_{1_{k-1}}^i$ ). Вычислить

$$e_{1_k \bullet}^i = e_{1_{k-1} \bullet}^i \cdot E_{1_k}^k, \quad i = 1, \dots, k-1.$$

Не нарушая общности, при анализе элементов  $x_i^k, e_{1_k \bullet}^i$  можно исключить из дальнейшего рассмотрения идентификацию следующих ситуаций:

- $x_i^k < 0, e_{1_k \bullet}^i \leq 0$  — несовместности ограничений задачи (1);
- $x_i^k = 0, e_{1_k \bullet}^i \leq 0$  или  $e_{1_k \bullet}^i \geq 0$  — уменьшения размерности задачи (1) и, как следствие, числа ненулевых элементов главных строк мультипликаторов, полученных на предыдущих шагах.

**Шаг 6** (пересчет  $k$ ). Положить  $k = k + 1$ . Если  $k \neq m$ , то перейти к шагу 1.

**Шаг 7** (расчет  $x^*$ ). Если  $x^m \geq 0$ , то вычислить  $x^m \Rightarrow x^*$  и остановиться; иначе перейти к шагу 1 второго этапа решения задачи:

$$\begin{aligned} & \min_{x \in R^n} (c_{m+1}^m x_{m+1} + c_{m+2}^m x_{m+2} + \dots + c_n^m x_n), \\ & x_i = x_i^m + e_{1_{m+1}m}^i x_{m+1} + e_{1_{m+2}m}^i x_{m+2} + \dots + e_{1_n m}^i x_n \geq 0, \quad i = 1, \dots, m, \\ & x_i \geq 0, \quad i = m+1, \dots, n. \end{aligned} \quad (5)$$

Ниже приводится детальное описание всех операций, выполняемых по ходу решения (5). Если в процессе вычислений элементы становятся меньше по абсолютной величине, так называемого критического значения  $\varepsilon_0$ , то их предлагается приравнять нулю.

## 2.2. Второй этап алгоритма

**Шаг 0** (инициализация). Положить

$$v = 0$$

и переписать (5) в виде

$$\begin{aligned} & \min_{x_v \in R^n} (c_{v_{m+1}}^v x_{v_{m+1}} + c_{v_{m+2}}^v x_{v_{m+2}} + \dots + c_{v_n}^v x_{v_n}), \\ & x_{v_i} = x_{v_i}^v + e_{v_{m+1}v}^i x_{v_{m+1}} + e_{v_{m+2}v}^i x_{v_{m+2}} + \dots + e_{v_n v}^i x_{v_n} \geq 0, \quad i = 1, \dots, m, \\ & x_{v_i} \geq 0, \quad i = m+1, \dots, n. \end{aligned} \quad (6)$$

**Шаг 1** (расчет  $v_q, v_r$ ). Вычислить номер  $v_q$  ведущего уравнения связи по формуле

$$\theta_{v_q} = \min_i x_{v_i}^v, \quad x_{v_i}^v < 0, \quad i = 1, \dots, m.$$

Вычислить номер  $v_r$  ведущего столбца по формуле

$$\Theta_{v_r} = \min_i \frac{c_{v_i}^v}{e_{v_q i}^v}, \quad e_{v_q i}^v > 0, \quad i = m+1, \dots, n.$$

**Шаг 2** (пересчет  $\nu$ ). Положить  $\nu = \nu + 1$ .

**Шаг 3** (расчет  $x_{\nu}^y$ ). Вычислить

$$x_{\nu_r}^y = -\frac{x_{\nu-1_q}^{y-1}}{e_{\nu-1_qr}^{y-1}}, \quad x_{\nu_i}^y = x_{\nu-1_i}^{y-1} + e_{\nu-1_i r}^{y-1} x_{\nu_r}^y, \quad i \neq r.$$

Положить

$$x_{\nu}^y = (x_{\nu_1}^y \quad x_{\nu_2}^y \quad \cdots \quad x_{\nu_m}^y \quad 0 \quad \cdots \quad 0)^T.$$

Если  $x_{\nu}^y \geq 0$ , то вычислить  $x_{\nu}^y \Rightarrow x^*$  и остановиться.

**Шаг 4** (расчет  $e_{\nu_i \bullet}^y, c_{\nu}^y$ ). Переписать уравнение связи переменной  $x_{\nu_q} \geq 0$ :

$$\begin{aligned} x_{\nu_r} &= x_{\nu_r}^y + e_{\nu_r m+1}^y x_{\nu_{m+1}} + e_{\nu_r m+2}^y x_{\nu_{m+2}} + \cdots + e_{\nu_r n}^y x_{\nu_n}, \\ x_{\nu_r}^y &= -\frac{x_{\nu-1_q}^{y-1}}{e_{\nu-1_qr}^{y-1}}, \quad e_{\nu_r r}^y = \frac{1}{e_{\nu-1_qr}^{y-1}}, \quad e_{\nu_r i}^y = -\frac{e_{\nu-1_qi}^{y-1}}{e_{\nu-1_qr}^{y-1}}, \quad i \neq r, \\ E_{\nu_r}^y &= \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ e_{\nu_r m+1}^y & \cdots & e_{\nu_r r}^y & \cdots & e_{\nu_r n}^y \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}. \end{aligned}$$

Вычислить

$$\begin{aligned} e_{\nu_i \bullet}^y &= e_{\nu-1_i}^{y-1} E_{\nu_r}^y, \quad i \neq q, \\ c_{\nu}^y &= (c_{\nu_{m+1}}^y \quad c_{\nu_{m+2}}^y \quad \cdots \quad c_{\nu_n}^y) = c_{\nu-1}^{y-1} E_{\nu_r}^y \end{aligned}$$

и перейти к шагу 1. Без ограничения общности можно исключить из дальнейшего рассмотрения анализ элементов  $x_{\nu_i}^y, e_{\nu_i \bullet}^y$  для идентификации следующих ситуаций:

- $x_{\nu_i}^y < 0, e_{\nu_i \bullet}^y \leq 0$  — несовместности ограничений задачи (6);
- $x_{\nu_i}^y = 0, e_{\nu_i \bullet}^y \leq 0$  или  $e_{\nu_i \bullet}^y \geq 0$  — уменьшения размерности задачи (6).

**ЗАМЕЧАНИЕ.**  $E_{\nu_r}^y$  —  $(n-m) \times (n-m)$ -матрица,  $\nu_r$ -я строка которой произвольная, а остальные — строки единичной  $(n-m) \times (n-m)$ -матрицы.  $E_{\nu_r}^y$  совпадает с мультипликатором в алгоритме симплекс-метода с точностью до знака транспонирования [Хакимова, Зеленков, Рзун, 2010].

### 2.3. Формы записи задачи, решаемой на втором этапе алгоритма

Обозначим:

$$\nu = n - m,$$

$0_{m\nu}$  —  $(m \times \nu)$ -матрица с нулевыми элементами,

$I_s$  — единичная  $(s \times s)$ -матрица,

$0_s$  — нулевой  $s$ -мерный вектор,

$$c' = (c_{\nu_{m+1}}^y \quad c_{\nu_{m+2}}^y \quad \cdots \quad c_{\nu_n}^y) — \nu\text{-мерная вектор-строка,}$$

$$c'' = (0_m^T \quad c') — n\text{-мерная вектор-строка в блочном виде,}$$

$$b = (x_{\nu_1}^y \quad x_{\nu_2}^y \quad \cdots \quad x_{\nu_m}^y)^T — m\text{-мерный вектор,}$$

$$\bar{z} = (b^T \quad 0_{\nu}^T)^T — n\text{-мерный вектор в блочном виде,}$$

$$z = (x_{\nu_1} \quad x_{\nu_2} \quad \cdots \quad x_{\nu_n})^T — n\text{-мерный вектор переменных,}$$

$y = (z_{m+1} \ z_{m+2} \ \dots \ z_n)^T$  —  $\nu$ -мерный вектор переменных,

$N = \begin{pmatrix} e_{v_1 m+1}^v & \dots & e_{v_1 n}^v \\ \vdots & \ddots & \vdots \\ e_{v_m m+1}^v & \dots & e_{v_m n}^v \end{pmatrix}$  —  $(m \times \nu)$ -матрица ранга  $m$ ,

$A = \begin{pmatrix} I_m & -N \end{pmatrix}$  —  $(m \times n)$ -матрица ранга  $m$  в блочном виде,

$K = \begin{pmatrix} -N^T & -I_\nu \end{pmatrix}$  —  $(\nu \times n)$ -матрица в блочном виде.

При введенных обозначениях задача (6) принимает вид

$$\min_{z \in R^n} (c'y), \quad z = \bar{z} - K^T y \geq 0, \tag{7}$$

где  $z = \bar{z} - K^T y$  — общее решение неоднородной системы линейных уравнений  $Az = b$ ,  $\bar{z}$  — частное решение системы,  $K^T y$  — общее решение однородной системы  $Az = 0$ .

Задачу (7) можно записать в виде стандартной задачи ЛП с ограничениями только типа неравенств:

$$\min_{y \in R^\nu} (c'y), \quad \bar{z} - K^T y \geq 0. \tag{8}$$

Запись (8) иногда называют задачей с односторонними условиями или сопряженной канонической формой [Моисеев, Иванчиков, Столярова, 1978]. С математической точки зрения задача (8) эквивалентна задаче ЛП в канонической форме:

$$\min_{z \in R^n} (c''z), \quad Az = b, \quad z \geq 0. \tag{9}$$

Обозначим:

$\ker A$  — нуль-пространство (ядро)  $(m \times n)$ -матрицы  $A$  ранга  $m$ :

$$\ker A = \{z \in R^n \mid Az = 0\},$$

$\text{im } A$  — пространство строк матрицы  $A$  (образ матрицы  $A^T$ ):

$$\text{im } A^T = \{\xi \in R^n \mid \xi = Ay, \ y \in R^m\}.$$

Размерность линейного пространства  $\ker A$  равна  $\nu = n - m$  — дефекту матрицы  $A$ . Нуль-пространство и пространство строк матрицы  $A$  являются ортогональными дополнениями друг к другу. Пространство  $R^n$  разлагается в прямую сумму этих подпространств:

$$R^n = \text{im } A^T \oplus \ker A.$$

Строки матрицы  $K$  линейно независимы по построению, принадлежат нуль-пространству матрицы  $A$ , и поэтому натянутое на них пространство  $\text{im } A^T$  совпадает с нуль-пространством (ядром) матрицы  $A$ . Таким образом,  $\text{im } K^T$  является ортогональным дополнением к пространству  $\text{im } A^T$ . Поэтому

$$\text{im } K^T = \ker A, \quad AK^T = 0_{m\nu}, \quad R^n = \text{im } A^T \oplus \ker A.$$

Покажем, что, по существу, (8) и (9) — это одна и та же задача. Для этого, следуя Голикову и Евтушенко [Голиков, Евтушенко, 2000], для  $(n \times \nu)$ -матрицы  $K^T$  определим псевдообратную матрицу  $(K^T)^+$  и определим множества

$$Y = \{y \in R^\nu \mid \bar{z} - K^T y \geq 0\},$$

$$Z = \{z \in R^n \mid Az = b, \ z \geq 0\},$$

$$\bar{Z} = \{z \in R^n \mid Az = b\}.$$

Формулу

$$z = \bar{z} - K^T y \quad (10)$$

можно рассматривать как аффинное отображение из  $R^v$  в  $R^n$ . При этом образом множества  $Y$  является множество  $Z$ . Между  $Z$  и  $Y$  существует взаимно однозначное соответствие. Действительно, для любого  $y \in Y$  по формуле (10) однозначно определяется  $z \in Z$ . Для переопределенной системы (10) полного ранга, содержащей  $n$  линейных уравнений и  $v$  неизвестных  $y$ , всегда определено псевдорешение

$$y(z) = (KK^T)^{-1} K(\bar{z} - z) = (K^T)^+ (\bar{z} - z), \quad (11)$$

которое является единственным решением системы (8) тогда и только тогда, когда  $\bar{z} - z \in \text{im } K^T$ . Это включение имеет место тогда и только тогда, когда  $z \in \bar{Z}$ . Итак, для любого  $z \in \bar{Z}$  формула (11) определяет аффинное преобразование, обратное к (10). Поэтому можно записать

$$Y = (K^T)^+ (\bar{z} - Z). \quad (12)$$

По существу, (8), (9) — это одна и та же задача. Внешнее отличие в записи связано с заменой переменных (10), позволившей по формуле (12) преобразовать множество  $Z$  в множество  $Y$ , являющееся пересечением  $n$  полупространств.

#### 2.4. Геометрическая интерпретация

Сначала, следуя Моисееву, Иванилу и Столяровой [Моисеев, Иванилов, Столярова, 1978], опишем геометрическую интерпретацию задач вида (9), затем — геометрическое представление итераций симплекс-метода и прямого алгоритма.

Введем для (9) дополнительную переменную:

$$z_0 = c'_1 z_1 + c'_2 z_2 + \dots + c'_n z_n,$$

запишем ее в следующем виде:

$$\min(z_0), \quad (13)$$

$$c'_1 z_1 + c'_2 z_2 + \dots + c'_n z_n = z_0, \quad (14)$$

$$a_{i1} z_1 + a_{i2} z_2 + \dots + a_{in} z_n = b_i, \quad i = 1, \dots, m,$$

$$z_j \geq 0, \quad j = 1, \dots, n. \quad (15)$$

Правая часть уравнений (13) представляет собой вектор

$$\tilde{b}^j = (z_0 \quad b_1 \quad \dots \quad b_m)^T$$

с  $m$  фиксированными компонентами и одной переменной компонентой  $z_0$ . Левая часть является линейной комбинацией расширенных векторов условий

$$\tilde{a}^j = (c'_j \quad b_{1j} \quad \dots \quad a_{mj})^T, \quad j = 1, \dots, n,$$

с неотрицательными коэффициентами  $z_j$ . Рассмотрим множество  $(m+1)$ -мерных векторов

$$u = (u_0 \quad u_1 \quad \dots \quad u_m)^T,$$

компоненты которых определены соотношениями

$$\begin{aligned} u_0 &= c'_1 z_1 + c'_2 z_2 + \dots + c'_n z_n, \\ u_i &= a_{i1} z_1 + a_{i2} z_2 + \dots + a_{in} z_n, \quad i = 1, \dots, m, \\ z_j &\geq 0, \quad j = 1, \dots, n. \end{aligned}$$

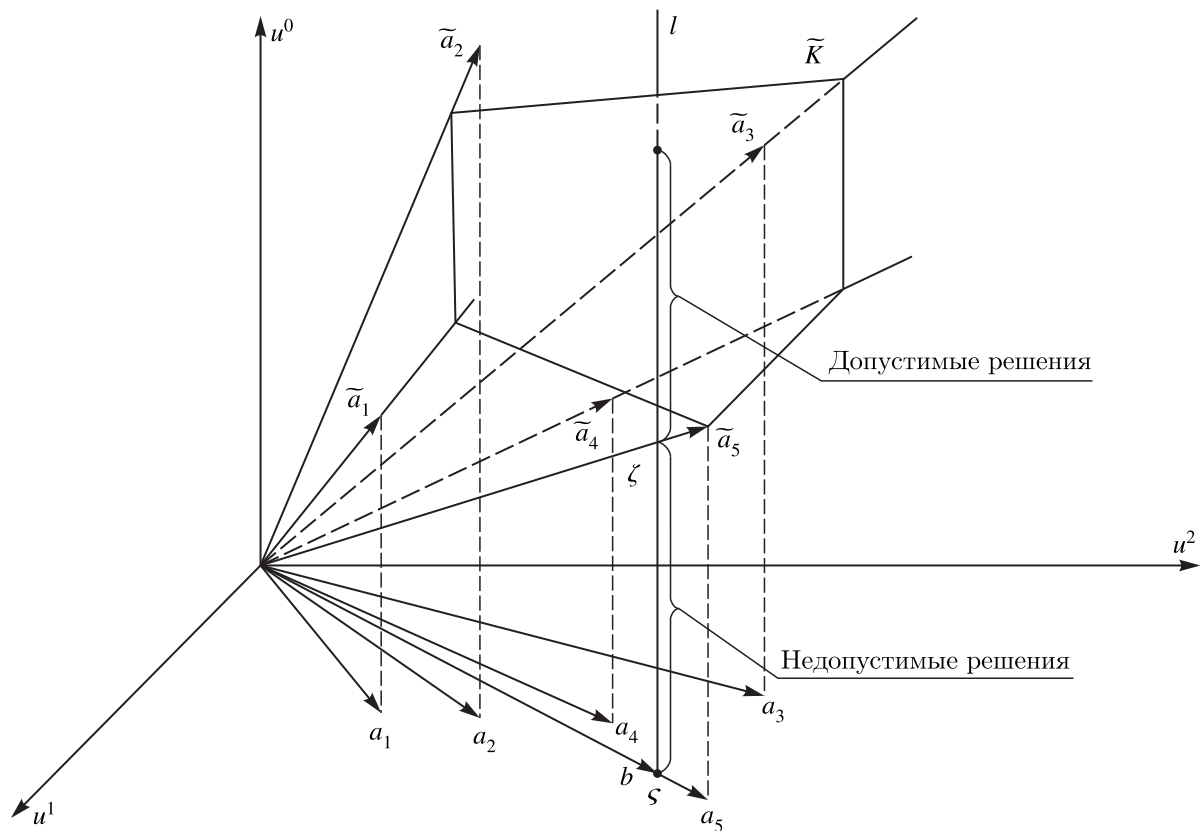


Рис. 1. Геометрическая интерпретация

На рис. 1 это множество показано для случая, когда  $m = 2$ , т. е. задача имеет два ограничения типа равенства.

Неотрицательные линейные комбинации расширенных векторов-условий образуют многогранный конус  $\tilde{K}$ , ребрами которого будут векторы  $\tilde{a}^1, \tilde{a}^2, \dots, \tilde{a}^5$ . Проекции их на плоскость  $u_0 = 0$  есть  $a^1, a^2, \dots, a^5$  соответственно (столбцы матрицы условий  $A$ ). Длина перпендикуляра, соединяющего конец вектора  $\tilde{a}^j$  с концом его проекции  $a^j$ , равна коэффициенту линейной формы  $c''_j$ .

При произвольном значении  $z_0$  вектор  $\tilde{b}$  указывает в некоторую точку, лежащую на вертикальной прямой  $l$  (см. рис. 1), проходящей через конец вектора

$$b = (b_1 \quad b_2 \quad \dots \quad b_m)^T$$

на плоскости  $u_0 = 0$  (точку  $\zeta$  с координатами  $(0 \quad b_1 \quad \dots \quad b_m)^T$  на рис. 1). Если же  $z_0$  — компонента какого-нибудь допустимого решения задачи (13)–(15), вектор  $\tilde{b}$  будет принадлежать также конусу  $\tilde{K}$ . Очевидно и то, что, если вектор  $\tilde{b}$  указывает в точку пересечения конуса  $\tilde{K}$  и прямой  $l$ , соответствующее значение  $z_0$  будет компонентой некоторого допустимого решения. Таким образом, допустимое множество решений задачи (13)–(15) отображается на отрезок прямой  $l$ , принадлежащий конусу  $\tilde{K}$ . Если прямая  $l$  проходит вне этого конуса, задача не имеет допустимых решений. Когда конус  $\tilde{K}$  содержит ось  $u_0$ , значение критерия на допустимом множестве не ограничено снизу, однако, по построению задачи (7), это невозможно.

Значения целевой функции задачи (13)–(15) на любом из ее допустимых решений, соответствующих определенному вектору  $\tilde{b}$ , совпадают и равны его первой координате. Оптимальные





**Шаг 2.** Определить ведущий столбец, столбец для ввода в базис:

$$d_k = \min \{d_k^1, d_k^2\},$$

$$d_k^1 = \min_i d_i^1 = \min_i (u^v a_{\bullet i} - c_i), \quad (17)$$

$$d_k^2 = \min_j d_j^2 = \min_j u_j^v. \quad (18)$$

Если  $d_k \geq \varepsilon_{\text{opt}}$ , то оптимум достигнут, где  $\varepsilon_{\text{opt}}$  — минимально допустимое значение ведущего элемента, выбираемое из соображений стабильности вычислительного процесса.

**Шаг 3.** Разложить ведущий столбец по формуле

$$X_k = B_v^{-1} a_{\bullet k},$$

если номер  $k$  определялся из (17), и по формуле

$$X_k = B_v^{-1} E_k,$$

если номер  $k$  определялся из (18). Здесь  $E_k$  —  $k$ -й столбец единичной матрицы  $E$ .

**Шаг 4.** Вычислить номер ведущей строки из соотношения

$$\theta = \frac{\bar{x}_r}{x_{rk}} = \min_j \left\{ \frac{\bar{x}_j}{x_{jk}}, x_{jk} > \varepsilon_{\text{piv}} \right\},$$

где  $\varepsilon_{\text{piv}}$  — минимально допустимое значение ведущего элемента, выбираемое из соображений стабильности вычислительного процесса.

**Шаг 5.** Пересчитать базисное решение:

$$\bar{x}'_j = \bar{x}_j - \theta x_{jk}, \quad j \neq r, \quad \bar{x}'_r = \theta,$$

пересчитать обратную базисную матрицу по формуле

$$B_{v+1}^{-1} = E_{r_{v+1}}^{v+1} B_v^{-1},$$

где  $E_{r_{v+1}}^{v+1}$  — элементарная матрица, мультипликатор:

$$E_{r_{v+1}}^{v+1} = \begin{pmatrix} 1 & & -\frac{x_{1k}}{x_{rk}} & & \\ & \ddots & \vdots & & \\ & & \frac{1}{x_{rk}} & & \\ & & \vdots & \ddots & \\ & & -\frac{x_{mk}}{x_{rk}} & & 1 \end{pmatrix}.$$

Шаги с 1 по 5 повторяются до тех пор, пока не будет достигнут оптимум или пока на шаге 4 не выяснится, что  $x_{jk} \leq \varepsilon_{\text{piv}}$ ,  $j = 1, \dots, m$ , иными словами, решение задачи не ограничено.

Симплекс-метод решения задач ЛП, использующий мультипликативное представление обратной базисной матрицы, как алгоритм решения системы уравнений, математически привлекателен, но уязвим в вычислительном отношении. В качестве примера рассмотрим систему, состоящую из одного уравнения:  $7x = 21$ . Лучший способ решения этой системы — деление:  $x = \frac{21}{7} = 3$ , а использование обратной матрицы приводит к вычислению  $x = (7^{-1})21 = 2,99997$ , которое требует больше арифметических операций и дает менее точный результат. Все сказанное справедливо и для систем со многими уравнениями. Лишние действия — главная причина, по которой в данной работе основное внимание уделяется прямым мультипликативным методам решения систем, а не способам представления обратной базисной матрицы.

При построении прямого алгоритма в терминах симплекс-метода будем считать, что на каждой итерации алгоритма к последовательности мультипликаторов добавляется еще один и что размер этой последовательности неудобно велик, но точно неизвестен.



Если  $\theta \leq \varepsilon_{\text{piv}}$ ,  $x_{\bar{r}k} \leq \varepsilon_{\text{piv}}$ , то условия задачи несовместны; если  $\theta \leq \varepsilon_{\text{piv}}$ ,  $x_{\bar{r}k} > \varepsilon_{\text{piv}}$ , то положить  $r = \bar{r}$  и перейти к шагу 4; если  $\theta > \varepsilon_{\text{piv}}$ , то перейти к шагу 4.

**Шаг 4.** Пересчитать текущее двойственное решение вектора оценок строк:

$$u^{v+1} = u^v - \frac{d_k}{x_{\bar{r}k}} (E_r)^T B_v^{-1},$$

где  $E_r$  —  $r$ -й столбец единичной матрицы.

**Шаг 5.** Пересчитать обратную базисную матрицу:

$$B_{v+1}^{-1} = E_{r_{v+1}}^{v+1} B_v^{-1},$$

где  $E_{r_{v+1}}^{v+1}$  — элементарная матрица, мультипликатор:

$$E_{r_{v+1}}^{v+1} = \begin{pmatrix} 1 & & -\frac{x_{1k}}{x_{rk}} & & \\ & \ddots & \vdots & & \\ & & \frac{1}{x_{rk}} & & \\ & & \vdots & \ddots & \\ & & -\frac{x_{mk}}{x_{rk}} & & 1 \end{pmatrix}.$$

**Шаг 6.** Если  $r \neq \bar{r}$ , то положить  $v = v + 1$  и перейти к шагу 1 первого этапа; иначе вычислить базисное решение:

$$\bar{x}_j = -\frac{x_{jk}}{x_{rk}}, \quad j \neq r, \quad \bar{x}_r = \frac{1}{x_{rk}},$$

положить  $v = v + 1$  и перейти к шагу 1 второго этапа.

Шаги с 1 по 6 повторяются до тех пор, пока не выяснится, что:

- решение задачи неограниченно (шаг 1);
- условия задачи несовместны (шаг 3);
- выполнено условие перехода к шагу 1 второго этапа (шаг 6).

### 3.3. Второй этап прямого алгоритма

**Шаг 1.** Определить ведущий столбец, столбец для ввода в базис:

$$d_k = \min \{d_k^1, d_k^2\},$$

$$d_k^1 = \min_i d_i^1 = \min_i (u^v a_i - c_i), \tag{22}$$

$$d_k^2 = \min_j d_j^2 = \min_j u_j^v. \tag{23}$$

Если  $d_k \geq \varepsilon_{\text{opt}}$ , то оптимум достигнут.

**Шаг 2.** Разложить ведущий столбец по формуле

$$X_k = B_v^{-1} a_{\bullet k},$$

если номер  $k$  определялся из (22), и по формуле

$$X_k = B_v^{-1} E_k,$$

если номер  $k$  определялся из (23).



В первом случае

$$\begin{aligned} c_r^y - \frac{d_k}{x_{rk}} &= c_r^y - \frac{u^y a_{\bullet k} - c_k^y}{x_{rk}} = c_r^y - \frac{c^y B_v^{-1} a_{\bullet k} - c_k^y}{x_{rk}} = \\ &= c_r^y - \frac{c^y (x_{1k} \ \cdots \ x_{mk})^T - c_k^y}{x_{rk}} = - \sum_{j=1, j \neq r}^m \frac{c_j^{y+1} x_{jk}}{x_{rk}} + \frac{c_r^{y+1}}{x_{rk}}. \end{aligned}$$

Во втором случае

$$\begin{aligned} c_r^y - \frac{d_k}{x_{rk}} &= c_r^y - \frac{u_k^y}{x_{rk}} = c_r^y - \frac{u^y E_k}{x_{rk}} = c_r^y - \frac{c^y B_v^{-1} E_k}{x_{rk}} = \\ &= c_r^y - \frac{c^y (x_{1k} \ \cdots \ x_{mk})^T}{x_{rk}} = - \sum_{j=1, j \neq r}^m \frac{c_j^y x_{jk}}{x_{rk}} = - \sum_{j=1, j \neq r}^m \frac{c_j^{y+1} x_{jk}}{x_{rk}} + \frac{c_r^{y+1}}{x_{rk}}. \end{aligned}$$

Утверждение доказано. □

Отсюда, в частности, следует, что прямой алгоритм — это алгоритм типа симплекс-метода.

При сравнении алгоритмов предположим, что обратная матрица  $B_v^{-1}$  заполнена целиком. Очевидно, что различие в числе арифметических операций возникает на шаге 1 алгоритма симплекс-метода и шаге 4 прямого алгоритма. Отсюда следует, что выигрыш в числе арифметических операций умножений и сложений на одной итерации равен  $m^2 - m - 1$ .

### 3.5. Число итераций

Рассмотрим задачу (6) в виде задачи ЛП с ограничениями только типа неравенств:

$$\min_{y \in R^v} (c' y), \quad b + Ny \geq 0, \quad y \geq 0. \tag{24}$$

В силу построения (6), задачи вида (24) удовлетворяют следующим условиям:

- $c' \geq 0, \nexists b \not\geq 0$ ;
- строки в матрицах  $N, (b \ N)$  линейно независимы по построению уравнений связи  $z_i = b_i + e_{v_i}^y \cdot y$  переменных  $z_i \geq 0$ , а строки в матрице в  $N' = \begin{pmatrix} c' \\ N \end{pmatrix}$  — по предположению о единственности решения задачи (1), равносильной (6);
- не имеется  $\nu$  или менее линейно зависимых строк в матрице  $\begin{pmatrix} N' \\ I_\nu \end{pmatrix}$  или  $\nu + 1$  или меньше линейно зависимых строк в  $\begin{pmatrix} b \ N \\ 0_\nu \ I_\nu \end{pmatrix}$ .

В частности, ЛП-задача в этой модели будет невырожденной. Обозначим:

$$\begin{aligned} p &= -b^T - m\text{-мерная вектор-строка,} \\ G &= (N \ -I_m)^T - (n \times m)\text{-матрица ранга } m \text{ в блочном виде,} \\ g &= (c' \ 0_m^T)^T - n\text{-мерный вектор в блочном виде.} \end{aligned}$$

При введенных обозначениях задача, двойственная (24), принимает вид

$$\max_{u \in R^m} (pu), \quad Gu \leq g. \tag{25}$$

Ограничимся выводом верхней границы  $\frac{m}{2}$  для задач вида (25), что согласуется с почти линейным ростом числа итераций симплекс-метода на практике. Модель естественная. Наряду с линейной программой вида (25) задана начальная вершина  $u^v$  (начальное значение счетчика числа итераций  $v = 0$ ). Затем выберем вектор-строку  $\bar{p}$ , удовлетворяющую условию

$$\bar{p}u^v = \max_{u \in R^m}(\bar{p}u), \quad Gu \leq g. \quad (26)$$

ЗАМЕЧАНИЕ. Для произвольной вершины  $u^v$  легко найти некоторую удовлетворяющую условию (26) вектор-строку  $\bar{p}$ , например, можно сложить строки  $G$ , в которых достигается равенство в системе  $Gu^v \leq g$  [Схрейвер, 1991]. Функция  $u \rightarrow \bar{p}u$  называется коцелевой функцией.

Затем выбирается одно из правил вычисления ведущей строки: Данцига или по наибольшему вкладу в изменение значения целевой функции. Первое правило проще, однако выбор второго при решении плохо обусловленных задач приводит к увеличению точности решения, а при возникновении последовательностей итераций, на которых целевая функция практически не изменяется, — к существенному уменьшению числа итераций. Таким образом, механизм выбора ведущей строки можно описать следующим образом. При решении плохо обусловленных задач выбирается второе правило, иначе — первое. Если выбрано второе правило, то механизм выбора описан. Иначе выбор ведущей строки обеспечивается внутренней структурой алгоритма так. На каждой итерации алгоритма оценивается приращение целевой функции. Если оно меньше заданного порогового значения, то выбирается второе правило, иначе — первое.

Прямой алгоритм решения невырожденной задачи (24) и алгоритм симплекс-метода решения (25) генерируют (из утверждения 1) одинаковую последовательность точек (базисное решение и текущее двойственное решение вектора оценок строк). Более того, геометрически итерации прямого алгоритма представляют собой переход от одной точки к другой в направлении возрастания целевой функции. Таким образом, механизм выбора ведущей строки удовлетворяет следующим условиям:

- |  |   |      |
|--|---|------|
| <ul style="list-style-type: none"> <li>(i) задача (25) невырожденная;</li> <li>(ii) геометрически итерации алгоритма симплекс-метода представляют собой переход от одной точки <math>u^v</math> к другой <math>u^{v+1}</math> в направлении возрастания целевой функции <math>u \rightarrow pu</math>, следовательно, и в направлении убывания коцелевой функции <math>u \rightarrow \bar{p}u</math>;</li> <li>(iii) для любого фиксированного <math>\bar{p}</math> обращение знаков некоторых неравенств, в которых достигается равенство в системе <math>Gu^v \leq g</math> (так, чтобы максимум <math>\bar{p}u</math> на измененном полиэдре был конечен), или замена <math>p</math> на <math>-p</math> не зависит от <math>G, g, p, \bar{p}</math>.</li> </ul> | } | (27) |
|--|---|------|

Практический смысл условий (27) можно поставить под сомнение, если не учитывать возникающие на практике плохо обусловленные задачи с сильной неустойчивостью численных решений относительно ошибок исходных данных или машинных округлений. Однако это прежде всего субъективная точка зрения автора.

ЗАМЕЧАНИЕ (ОБОСНОВАНИЕ). Не нарушая общности, будем считать, что на очередной итерации алгоритма решается система с матрицей Гильберта. Данные матрицы интересны тем, что они плохо обусловлены даже для сравнительно небольших значений их порядка. Эта особенность позволяет их использовать для проверки и отладки различных вычислительных алгоритмов. Системы с плохо обусловленными матрицами принято называть неустойчивыми или плохо обусловленными. В целом они характеризуются тем, что незначительное изменение условий счета может привести к недопустимо большим ошибкам в решении. Более того, известно, что применение математики «длинных чисел» позволяет получить основные характеристики матриц Гильберта (например, значения норм, чисел обусловленности и т. д.) до сотого порядка и выше [Майстренко, Светлаков, Черепанов, 2011]. Получить подобные результаты, используя

одинарную, двойную или максимальную машинную точность, невозможно в принципе. Таким образом, построение алгоритма решения подобных задач большой размерности, используя одинарную, двойную, максимальную машинную точность или математику «длинных чисел», также невозможно в принципе.

Теперь можно провести ведущую операцию и найти  $u^{v+1}$ .

Поскольку (из (27), (i))  $pu^{v+1} \neq pu^v$ ,  $\bar{p}u^{v+1} \neq \bar{p}u^v$  и (из (27), (ii))  $pu^{v+1} > pu^v$ ,  $\bar{p}u^{v+1} < \bar{p}u^v$  и программа (25) неограниченная, имеем

$$pu^{v+1} > pu^v, \quad \bar{p}u^{v+1} < \bar{p}u^v, \quad p\varphi = (u^{v+1} - u^v) > 0, \quad \bar{p}\varphi < 0. \quad (28)$$

**Утверждение 2.** Если механизм выбора ведущей строки удовлетворяет (27), то симплекс-метод решает задачи вида (25) за не более чем  $\frac{m}{2}$  итераций.

*Доказательство.* Пусть заданы задача (25), вершина  $u^v$  полиэдра  $P$  и вектор-строка  $\bar{p}$ , для которой  $u^v$  максимизирует  $\bar{p}u$  на множестве  $Gu \leq g$ . Пусть  $n$  и  $m$  — соответственно число строк и столбцов матрицы  $G$ .

Пусть  $\omega^1, \dots, \omega^t$  — векторы из  $R^n$ , являющиеся решениями подсистем из  $n$  линейно независимых уравнений системы  $Gu = g$ . Таким образом,  $t = C_n^m$ .

Рассмотрим класс  $L$  задач ЛП вида

$$\max_{u \in R^m}(pu), \quad \tilde{G}u \leq \tilde{g},$$

образованных из программы (25) путем обращения некоторого (возможно, пустого) множества неравенств и (или) замены вектора  $p$  на  $-p$ , для которых конечен максимум

$$\max_{u \in R^m}(\bar{p}u), \quad \tilde{G}u \leq \tilde{g}. \quad (29)$$

Покажем, что  $|L| = 2C_n^m$ . Действительно, из условия утверждения 1 вытекает, что каждый максимум (29) достигается на единственной вершине  $\tilde{G}u \leq \tilde{g}$  и, следовательно, на одном из векторов  $\omega^s$ . Более того, для каждого  $\omega^s$  существует ровно один выбор системы  $\tilde{G}u \leq \tilde{g}$ , при котором максимум (29) достигается на  $\omega^s$ . Поскольку ровно  $n$  неравенств из системы  $\tilde{G}u \leq \tilde{g}$ , допустим  $\tilde{G}'u \leq \tilde{g}'$ , выполняются на  $\omega^s$  как равенства. Таким образом, в остальных  $n-m$  неравенствах знак неравенства при подстановке  $\omega^s$  будет строгий и потому не может быть обращен. Более того, вектор-строка  $\bar{p}$  должна принадлежать конусу, порожденному строками  $\tilde{G}$ , и, следовательно, здесь знаки однозначно определены. Таким образом,  $\omega^s$  отвечают программы:

$$\max_{u \in R^m}(pu), \quad \tilde{G}u \leq \tilde{g}, \quad \max_{u \in R^m}(-pu), \quad \tilde{G}u \leq \tilde{g}.$$

Поэтому  $|L| = 2C_n^m$ .

Теперь рассмотрим множество всех конечных и бесконечных ребер всех полиэдров, заданных системами  $\tilde{G}u \leq \tilde{g}$ . Каждое множество из  $m-1$  уравнений системы  $Gu = g$  порождает прямую (одномерное аффинное подпространство) в  $R^m$  содержащую  $n-m+1$  из точек  $\omega^s$ , которые разбивают ее на  $n-m$  ребер-отрезков и два бесконечных ребра-луча. Пусть  $l$  — такая прямая, а  $\omega^s$  и  $\omega^\mu$  — две соседние точки на  $l$ , т. е. отрезок  $\overline{\omega^s \omega^\mu}$  не содержит других точек из  $\omega^1, \dots, \omega^t$ . Тогда ребро  $\overline{\omega^s \omega^\mu}$  проходится в одной из программ класса ровно один раз. Предположим противное, что  $\overline{\omega^s \omega^\mu}$  проходится в более чем одной программе:

$$\max_{u \in R^m}(\bar{p}u), \quad \tilde{G}u \leq \tilde{g}.$$

Для точки  $\omega^s$  найдем вектор-строку  $\bar{p}$ , удовлетворяющую условию

$$\bar{p}\omega^s = \max_{u \in R^m}(\bar{p}u), \quad u \in \bar{P} = \{u \mid \tilde{G}u \leq \tilde{g}\},$$

и для полиэдра  $\tilde{P}$  справедливо неравенство

$$\bar{p}u \leq \bar{p}\omega^s. \quad (30)$$

Из (27) вытекает, что ровно  $m-1$  из неравенств системы  $\tilde{G}u \leq \tilde{g}$  обращаются в равенства на  $\overline{\omega^s \omega^\mu}$ . Обозначим матрицу коэффициентов подсистемы таких неравенств ранга  $m-1$  через  $\tilde{G}_{eq(\overline{\omega^s \omega^\mu})}$ . Таким образом, знаки неравенства в других неравенствах системы нельзя обратить. Более того, существует ровно одна  $(m-1)$ -мерная вектор-строка  $w$ , удовлетворяющая  $w\tilde{G}_{eq(\overline{\omega^s \omega^\mu})} = \bar{p}$ . Поскольку неравенство (30) задает опорное подпространство, то  $w^T > 0$ . Из единственности  $w$  вытекает единственность знаков неравенства в  $\tilde{G}u \leq \tilde{g}$ . Наконец, если  $p\omega^s > p\omega^\mu$ , то из (28) следует  $\bar{p} = p$ . Если  $p\omega^s < p\omega^\mu$ , то из (28) вытекает, что  $\bar{p} = -p$ . Таким образом, для  $\bar{p}$  также имеется единственный выбор, так что ребро  $\overline{\omega^s \omega^\mu}$  проходится в единственной программе из  $L$ .

Аналогично из (28) следует, что каждый луч  $\omega^s + \varphi R_+$  проходится только в том случае, если  $\bar{p}\varphi > 0$ , и потому лишь в одной программе из  $L$ .

На каждой прямой, заданной  $m-1$  уравнениями из системы  $Gu = g$ , имеются  $n-m$  ребер и один луч  $\omega^s + \varphi R_+$  с  $\bar{p}\varphi > 0$ . Таким образом, полное число лучей и ребер равно  $(n-m+1)C_n^{m-1} = mC_n^m$ , поэтому число итераций не больше чем  $\frac{mC_n^m}{|L|} = \frac{1}{2}m$ .

Утверждение доказано.  $\square$

В частности, отсюда и из утверждения 1 следует, что прямой алгоритм решает задачи вида (24) за не более чем  $\frac{m}{2}$  итераций.

#### 4. Примеры решений задач с гарантированным поведением симплекс-метода

Рассмотрим прямой алгоритм на примерах решений задач с гарантированным поведением симплекс-метода в феноменологическом плане (в смысле понимания феномена) и конструктивном плане (как определяемое устроено и как с ним работать) [Лачинов, Поляков, 1999].

ПРИМЕР 1 (ШЕВЧЕНКО – ЗОЛОТЫХ [ШЕВЧЕНКО, ЗОЛОТЫХ, 2002]). Найти решение задачи (1) на заикливание симплекс-метода с правилом Данцига:

$$c = (0 \ 0 \ 0 \ 0 \ -200 \ -175 \ 1100 \ 2),$$

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & -3 & -\frac{5}{4} & 7 & \frac{1}{50} \\ -1 & 0 & 0 & 0 & -\frac{1}{3} & -\frac{1}{6} & 1 & \frac{1}{50} \\ 0 & 0 & 1 & 0 & \frac{75}{2} & -\frac{25}{4} & \frac{175}{2} & \frac{1}{4} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad h = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

Покажем, что решение задачи Шевченко – Золотых достигается на первом этапе алгоритма. В процессе решения участвуют следующие вспомогательные величины:

$c_0$  ( $-c_0 \leq cx^*$ ) – сколь угодно большая скалярная величина, вводится для уменьшения ошибок компенсации;

$x_0$  ( $x_0 \geq 0$ ) – дополнительная переменная;

$c_0 - x_0 + cx = 0$  – уравнение связи дополнительной переменной;

$\min_{x \in R^n}(x_0)$ ,  $c_0 - x_0 + cx = 0$ ,  $h + Hx = 0$ ,  $x_0 \geq 0$ ,  $x \geq 0$ , – задача, эквивалентная задаче Шевченко – Золотых.



ЗАМЕЧАНИЕ. Среди арифметических операций есть такие, которые могут приводить к появлению относительных ошибок, превышающих величину машинной точности во много раз, например вычисления разностей почти совпадающих округленных чисел. Связанные с ними ошибки принято называть ошибками компенсации [Гилл, Мюррей, Райт, 1985].

**Шаг 0** (инициализация). Вычислить номер  $r$  ведущего столбца (номер ведущей строки  $q = 0$ ) по формуле

$$\Theta_r = \min_{i=1, \dots, 8, c_i < 0} c_i = c_5 = -200.$$

Переписать уравнение связи дополнительной переменной  $x_0 \geq 0$ :

$$x_5 = x_5^0 + e_{5_0}^0 \cdot (x_1 \ x_2 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7 \ x_8)^T,$$

$$E_{5_0}^0 = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & -\frac{1}{200} & -\frac{7}{8} & \frac{11}{2} & \frac{1}{100} & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & & 1 \end{pmatrix}.$$

Положить

$$x^0 = (0 \ 0 \ 0 \ 0 \ \frac{1}{200}c_0 \ 0 \ 0 \ 0)^T.$$

**Шаг 1** (расчет  $x^1$ ). Вычислить элементы уравнения связи четвертой строки (выберем номер ведущей строки  $q = 4$ ):

$$h_4^1 + (h_{41}^1 \ \dots \ h_{48}^1)(x_1 \ x_2 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7 \ x_8)^T = 0,$$

$$h_4^1 = h_4 + h_{45}x_5^0 = 1, \quad (h_{41}^1 \ \dots \ h_{48}^1) = h_4 \cdot E_{5_0}^0 = (0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1).$$

Переписать уравнение связи четвертой строки (для упрощения вычислений выберем номер ведущего столбца  $r = 8$ ):

$$x_8 = x_8^1 + e_{8_1}^1 \cdot (x_1 \ x_2 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T,$$

$$x_8^1 = 1, \quad e_{8_1}^1 = (0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 0),$$

$$E_{8_1}^1 = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & -1 \end{pmatrix}.$$

Пересчитать элементы уравнения связи:

$$x_5 = x_5^1 + e_{5_1}^0 \cdot (x_1 \ x_2 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T,$$

$$x_5^1 = \frac{1}{100} + \frac{1}{200}c_0, \quad e_{5_1}^0 \cdot = e_{5_0}^0 \cdot E_{8_1}^1 = \left(0 \ 0 \ 0 \ -\frac{1}{100} \ -\frac{1}{200} \ -\frac{7}{8} \ \frac{11}{2}\right),$$

$$E_{5_0}^0 E_{8_1}^1 = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & -\frac{1}{100} & -\frac{1}{200} & -\frac{7}{8} & \frac{11}{2} \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & -1 \end{pmatrix}.$$

Положить

$$x^1 = \left(0 \ 0 \ 0 \ 0 \ \frac{1}{100} + \frac{1}{200}c_0 \ 0 \ 0 \ 1\right)^T.$$

**Шаг 2** (расчет  $x^2$ ). Вычислить элементы уравнения связи первой строки (выберем  $q = 1$ ):

$$h_1^2 + (h_{11}^2 \ h_{12}^2 \ \dots \ h_{17}^2)(x_1 \ x_2 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T = 0,$$

$$h_1^2 = h_1 + h_{15}x_5^1 + h_{18}x_8^1 = -\frac{1}{100} - \frac{3}{200}c_0,$$

$$(h_{11}^2 \ h_{12}^2 \ \dots \ h_{17}^2) = h_1 \cdot E_{5_0}^0 E_{8_1}^1 = \left(0 \ 1 \ 0 \ \frac{1}{100} \ \frac{3}{200} \ \frac{11}{8} \ -\frac{19}{2}\right).$$

Переписать уравнение связи первой строки (выберем  $r = 2$ ):

$$x_2 = x_2^2 + e_{2_2}^2 \cdot (x_1 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T,$$

$$x_2^2 = \frac{1}{100} + \frac{3}{200}c_0, \quad e_{2_2}^2 \cdot = \left(0 \ 0 \ -\frac{1}{100} \ -\frac{3}{200} \ -\frac{11}{8} \ \frac{19}{2}\right),$$

$$E_{2_2}^2 = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix}.$$

Пересчитать элементы уравнений связи:

$$x_8 = x_8^2 + e_{8_2}^1 \cdot (x_1 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T,$$

$$x_8^2 = 1, \quad e_{8_2}^1 \cdot = e_{8_1}^1 \cdot E_{2_2}^2 = \left(0 \ 0 \ -1 \ 0 \ 0 \ 0\right),$$

$$x_5 = x_5^2 + e_{5_2}^0 \cdot (x_1 \ x_3 \ x_4 \ x_0 \ x_6 \ x_7)^T,$$

$$x_5^2 = \frac{1}{100} + \frac{1}{200}c_0, \quad e_{5_2}^0 \cdot = e_{5_1}^0 \cdot E_{2_2}^2 = \left(0 \ 0 \ -\frac{1}{100} \ -\frac{1}{200} \ -\frac{7}{8} \ \frac{11}{2}\right),$$



$$E_{5_0}^0 E_{8_1}^1 E_{2_2}^2 E_{6_3}^3 = \begin{pmatrix} 1 & & & \\ -11 & -\frac{29}{150} & \frac{1}{300} & \frac{1}{3} \\ & 1 & & \\ -7 & -\frac{19}{150} & \frac{1}{150} & -\frac{1}{3} \\ 8 & -\frac{2}{15} & -\frac{2}{150} & \frac{20}{3} \\ & & & -1 \end{pmatrix}.$$

Положить

$$x^3 = \left( 0 \quad \frac{29}{150} - \frac{1}{300}c_0 \quad 0 \quad 0 \quad \frac{19}{150} - \frac{1}{150}c_0 \quad -\frac{2}{15} + \frac{2}{150}c_0 \quad 0 \quad 1 \right)^T.$$

**Шаг 4** (расчет  $x^4$ ). Вычислить элементы уравнения связи третьей строки:

$$\begin{aligned} h_3^4 + (h_{31}^4 \quad h_{32}^4 \quad \dots \quad h_{35}^4)(x_1 \quad x_3 \quad x_4 \quad x_0 \quad x_7)^T &= 0, \\ h_3^4 = h_3 + h_{32}x_2^3 + h_{35}x_5^3 + h_{36}x_6^3 + h_{38}x_8^3 &= \frac{35}{6} - \frac{1}{3}c_0, \\ (h_{31}^4 \quad h_{32}^4 \quad \dots \quad h_{35}^4) = h_3 \bullet E_{5_0}^0 E_{8_1}^1 E_{2_2}^2 E_{6_3}^3 &= \left( -\frac{625}{2} \quad 1 \quad -\frac{35}{6} \quad \frac{1}{3} \quad -\frac{325}{6} \right). \end{aligned}$$

Переписать уравнение связи третьей строки ( $r = 2$ ):

$$\begin{aligned} x_3 &= x_3^4 + e_{3_4}^4 \bullet (x_1 \quad x_4 \quad x_0 \quad x_7)^T, \\ x_3^4 &= -\frac{35}{6} + \frac{1}{3}c_0, \quad e_{3_4}^4 \bullet = \left( \frac{625}{2} \quad \frac{35}{6} \quad -\frac{1}{3} \quad \frac{325}{6} \right), \\ E_{3_4}^4 &= \begin{pmatrix} 1 & & & \\ \frac{625}{2} & \frac{35}{6} & -\frac{1}{3} & \frac{325}{6} \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}. \end{aligned}$$

Пересчитать элементы уравнений связи:

$$\begin{aligned} x_5 &= x_5^4 + e_{5_4}^0 \bullet (x_1 \quad x_4 \quad x_0 \quad x_7)^T, \\ x_5^4 &= \frac{19}{150} - \frac{1}{150}c_0, \quad e_{5_4}^0 \bullet = e_{5_3}^0 \bullet E_{3_4}^4 = \left( -7 \quad -\frac{19}{150} \quad \frac{1}{150} \quad -\frac{1}{3} \right), \\ x_6 &= x_6^4 + e_{6_4}^3 \bullet (x_1 \quad x_4 \quad x_0 \quad x_7)^T, \\ x_6^4 &= -\frac{2}{15} + \frac{1}{75}c_0, \quad e_{6_4}^3 \bullet = e_{6_3}^3 \bullet E_{3_4}^4 = \left( 8 \quad \frac{2}{15} \quad -\frac{2}{150} \quad \frac{20}{3} \right), \\ x_2 &= x_2^4 + e_{2_4}^2 \bullet (x_1 \quad x_4 \quad x_0 \quad x_7)^T, \\ x_2^4 &= \frac{29}{150} - \frac{1}{300}c_0, \quad e_{2_4}^2 \bullet = e_{2_3}^2 \bullet E_{3_4}^4 = \left( -11 \quad -\frac{29}{150} \quad \frac{1}{300} \quad \frac{1}{3} \right), \\ x_8 &= x_8^4 + e_{8_4}^1 \bullet (x_1 \quad x_4 \quad x_0 \quad x_7)^T, \\ x_8^4 &= 1, \quad e_{8_4}^1 \bullet = e_{8_3}^1 \bullet E_{3_4}^4 = \left( 0 \quad -1 \quad 0 \quad 0 \right). \end{aligned}$$

Переписать уравнение связи переменной  $x_5 \geq 0$  ( $x_5^4 < 0$ ):

$$\begin{aligned} x_0 &= x_0^4 + e_{0_4}^4 \cdot (x_1 \ x_4 \ x_5 \ x_7)^T, \\ x_0^4 &= -19 + c_0, \quad e_{0_4}^4 \cdot \bullet = (1050 \ 19 \ 150 \ 50), \\ E_{0_4}^4 &= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1050 & 19 & 150 & 50 \\ & & & 1 \end{pmatrix}. \end{aligned}$$

Пересчитать элементы уравнений связи:

$$\begin{aligned} x_2 &= x_2^5 + e_{2_5}^2 \cdot (x_1 \ x_4 \ x_5 \ x_7)^T, \\ x_2^5 &= \frac{13}{100}, \quad e_{2_5}^2 \cdot \bullet = e_{2_4}^2 \cdot E_{0_4}^4 = \left(-\frac{15}{2} \ -\frac{13}{100} \ \frac{1}{2} \ \frac{1}{2}\right), \\ x_3 &= x_3^5 + e_{3_5}^4 \cdot (x_1 \ x_4 \ x_5 \ x_7)^T, \\ x_3^5 &= \frac{1}{2}, \quad e_{3_5}^4 \cdot \bullet = e_{3_4}^4 \cdot E_{0_4}^4 = \left(-\frac{75}{2} \ -\frac{1}{2} \ -50 \ \frac{75}{2}\right), \\ x_6 &= x_6^5 + e_{6_5}^3 \cdot (x_1 \ x_4 \ x_5 \ x_7)^T, \\ x_6^5 &= \frac{3}{25}, \quad e_{6_5}^3 \cdot \bullet = e_{6_4}^3 \cdot E_{0_4}^4 = \left(-6 \ -\frac{3}{25} \ -2 \ 6\right), \\ x_8 &= x_8^5 + e_{8_5}^1 \cdot (x_1 \ x_4 \ x_5 \ x_7)^T, \\ x_8^5 &= 1, \quad e_{8_5}^1 \cdot \bullet = e_{8_4}^1 \cdot E_{0_4}^4 = (0 \ -1 \ 0 \ 0). \end{aligned}$$

Положить

$$x^* = x^4 = \left(0 \ \frac{13}{100} \ \frac{1}{2} \ 0 \ 0 \ \frac{3}{25} \ 0 \ 1\right)^T \geq 0, \quad cx^* = -19.$$

ЗАМЕЧАНИЕ. Ошибки компенсации минимизируются при исключении дополнительной переменной  $x_0$  в процессе расчета  $x^4$ .

ПРИМЕР 2 (ШЕВЧЕНКО – ЗОЛОТЫХ [ШЕВЧЕНКО, ЗОЛОТЫХ, 2002]). Найти решение задачи на заикливание симплекс-метода с правилом Данцига:

$$\begin{aligned} c &= (600 \ 0 \ 0 \ 0 \ 0 \ -75 \ 500 \ -2), \\ H &= \begin{pmatrix} 9 & 1 & 0 & 0 & 0 & \frac{1}{4} & -2 & -\frac{1}{25} \\ 3 & 0 & 1 & 0 & 0 & \frac{1}{2} & -3 & -\frac{1}{50} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -\frac{225}{2} & 0 & 0 & 0 & 1 & -25 & 200 & 1 \end{pmatrix}, \quad h = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \end{aligned}$$

Покажем, что решение задачи Шевченко – Золотых достигается на втором этапе алгоритма. Ниже будем использовать верхний индекс  $\nu$  как счетчик итераций. Начальное значение этого счетчика –  $\nu = 0$ , и задача, построенная на четвертом шаге первого этапа решения задачи Шевченко – Золотых [Свириденко, 2017], может быть переписана как

$$\min x_0 = e_{01}^\nu x_1 + e_{02}^\nu x_2 + e_{03}^\nu x_3 + e_{04}^\nu x_4,$$

$$\begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} x_5^y \\ x_6^y \\ x_7^y \\ x_8^y \end{pmatrix} + \begin{pmatrix} e_{5\bullet}^y \\ e_{6\bullet}^y \\ e_{7\bullet}^y \\ e_{8\bullet}^y \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geq 0,$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0,$$

$$e_{0\bullet}^y = (1925 \quad 100 \quad 100 \quad 8), \quad x^y = (0 \quad 0 \quad 0 \quad 0 \quad 3 \quad -\frac{8}{25} \quad -\frac{3}{50} \quad 1)^T,$$

$$e_{5\bullet}^y = (\frac{1825}{2} \quad -100 \quad 0 \quad -3), \quad e_{6\bullet}^y = (104 \quad 12 \quad -8 \quad \frac{8}{25}),$$

$$e_{7\bullet}^y = (-\frac{35}{2} \quad 2 \quad -1 \quad \frac{3}{50}), \quad e_{8\bullet}^y = (0 \quad 0 \quad 0 \quad -1).$$

**Шаг 1** (расчет  $x^1$ ). Вычислить номер  $q$  ведущего уравнения связи по формуле

$$\theta_q = \min_{i=5,6,7,8, x_i^0 < 0} x_i^0 = x_6^0 = -\frac{8}{25}.$$

Вычислить номер  $r$  ведущего столбца по формуле

$$\Theta_r = \min_{i=1,2,3,4, e_{6i}^0 > 0} \frac{e_{0i}^0}{e_{6i}^0} = \frac{e_{02}^0}{e_{62}^0} = \frac{25}{3}.$$

Переписать уравнение связи переменной  $x_6 \geq 0$  ( $x_6^0 < 0$ ):

$$x_2 = x_2^1 + e_{2\bullet}^1 (x_1 \quad x_6 \quad x_3 \quad x_4)^T,$$

$$x_2^1 = \frac{2}{75}, \quad e_{2\bullet}^1 = (-\frac{26}{3} \quad \frac{1}{12} \quad \frac{2}{3} \quad -\frac{2}{75}),$$

$$E_2^1 = \begin{pmatrix} 1 & & & \\ -\frac{26}{3} & \frac{1}{12} & \frac{2}{3} & -\frac{2}{75} \\ & & 1 & \\ & & & 1 \end{pmatrix}.$$

Пересчитать элементы уравнений связи:

$$x_5 = x_5^1 + e_{5\bullet}^1 (x_1 \quad x_6 \quad x_3 \quad x_4)^T,$$

$$x_5^1 = \frac{1}{3}, \quad e_{5\bullet}^1 = e_{5\bullet}^0 E_2^1 = (-\frac{275}{6} \quad \frac{25}{3} \quad -\frac{200}{3} \quad -\frac{1}{3}),$$

$$x_7 = x_7^1 + e_{7\bullet}^1 (x_1 \quad x_6 \quad x_3 \quad x_4)^T,$$

$$x_7^1 = -\frac{1}{150}, \quad e_{7\bullet}^1 = e_{7\bullet}^0 E_2^1 = (-\frac{209}{6} \quad \frac{1}{6} \quad \frac{1}{3} \quad \frac{1}{150}),$$

$$x_8 = x_8^1 + e_{8\bullet}^1 (x_1 \quad x_6 \quad x_3 \quad x_4)^T,$$

$$x_8^1 = 1, \quad e_{8\bullet}^1 = e_{8\bullet}^0 E_2^1 = (0 \quad 0 \quad 0 \quad -1).$$

Пересчитать элементы целевой функции:

$$e_{0\bullet}^1 = e_{0\bullet}^0 E_2^1 = (\frac{3175}{3} \quad \frac{25}{3} \quad \frac{500}{3} \quad \frac{16}{3}).$$

Положить

$$x^1 = (0 \quad \frac{2}{75} \quad 0 \quad 0 \quad \frac{1}{3} \quad 0 \quad -\frac{1}{150} \quad 1)^T.$$

**Шаг 2** (расчет  $x^2$ ). Вычислить номер  $q$  ведущего уравнения связи по формуле

$$\theta_q = \min_{i=2,5,7,8, x_i^1 < 0} x_i^1 = x_7^1 = -\frac{1}{150}.$$

Вычислить номер  $r$  ведущего столбца по формуле

$$\Theta_r = \min_{i=1,6,3,4, e_{7i}^0 > 0} \frac{e_{0i}^1}{e_{7i}^1} = \frac{e_{06}^1}{e_{76}^1} = 50.$$

Переписать уравнение связи переменной  $x_7 \geq 0$  ( $x_7^1 < 0$ ):

$$\begin{aligned} x_6 &= x_6^2 + e_{6\bullet}^2 \cdot (x_1 \quad x_7 \quad x_3 \quad x_4)^T, \\ x_6^2 &= \frac{1}{25}, \quad e_{6\bullet}^2 = (209 \quad 6 \quad -2 \quad -\frac{1}{25}), \\ E_6^2 &= \begin{pmatrix} 1 & & & \\ 209 & 6 & -2 & -\frac{1}{25} \\ & & 1 & \\ & & & 1 \end{pmatrix}. \end{aligned}$$

Пересчитать элементы уравнений связи:

$$\begin{aligned} x_2 &= x_2^2 + e_{2\bullet}^2 \cdot (x_1 \quad x_7 \quad x_3 \quad x_4)^T, \\ x_2^2 &= \frac{3}{100}, \quad e_{2\bullet}^2 = e_{2\bullet}^1 \cdot E_6^2 = \left(\frac{35}{4} \quad \frac{1}{2} \quad \frac{1}{2} \quad -\frac{3}{100}\right), \\ x_5 &= x_5^2 + e_{5\bullet}^2 \cdot (x_1 \quad x_7 \quad x_3 \quad x_4)^T, \\ x_5^2 &= 0, \quad e_{5\bullet}^2 = e_{5\bullet}^1 \cdot E_6^2 = \left(-\frac{3575}{2} \quad -50 \quad -50 \quad 0\right), \\ x_8 &= x_8^2 + e_{8\bullet}^2 \cdot (x_1 \quad x_7 \quad x_3 \quad x_4)^T, \\ x_8^2 &= 1, \quad e_{8\bullet}^2 = e_{8\bullet}^1 \cdot E_6^2 = (0 \quad 0 \quad 0 \quad -1). \end{aligned}$$

Пересчитать элементы целевой функции:

$$e_{0\bullet}^2 = e_{0\bullet}^1 \cdot E_6^2 = (2800 \quad 50 \quad 150 \quad 5).$$

Положить

$$x^* = x^2 = \left(0 \quad \frac{3}{100} \quad 0 \quad 0 \quad 0 \quad \frac{1}{25} \quad 0 \quad 1\right)^T \geq 0, \quad c^T x^* = -\frac{16}{3}.$$

**ПРИМЕР 3** (KLEE – SCHRIJVER [YAN, 2020]). Схрейвер доказал, что симплекс-метод решения задачи

$$\begin{aligned} \min_{x \in R^n} (cx), \quad h + Hx &\geq 0, \quad x \geq 0, \\ c &= (-2^{n-1} \quad -2^{n-2} \quad \dots \quad -2 \quad -1), \\ H &= \begin{pmatrix} -1 & & & & \\ -2^2 & -1 & & & \\ & & \ddots & & \\ -2^n & -2^{n-1} & \dots & -2^2 & -1 \end{pmatrix}, \quad h = \begin{pmatrix} 5 \\ 5^2 \\ \vdots \\ 5^n \end{pmatrix}, \end{aligned}$$

с правилом Данцига не полиномиален: число итераций алгоритма растет экспоненциально быстро [Схрейвер, 1991]. Допустимая область этой задачи является деформированным  $n$ -мерным кубом, так что симплекс-метод последовательно проходит все его  $2^n$  вершин. Причина этого состоит в комбинаторном характере симплекс-метода, последовательно перебирающего вершины многогранника допустимых решений при поиске оптимального решения. Таким образом, данный метод эффективен на небольшом наборе входных данных, при увеличении же их сложность алгоритма будет возрастать скачкообразно.

ЗАМЕЧАНИЕ.  $x^* = (0 \ \dots \ 0 \ 5^n)^T$  — единственное оптимальное решение задачи Klee–Schrijver [Схрейвер, 1991].

Покажем, что прямой алгоритм с правилом Данцига находит решение задачи Klee–Schrijver на первом шаге первого этапа. В процессе решения участвуют следующие вспомогательные величины:

$c_0$  ( $-c_0 \leq cx^*$ ) — сколь угодно большая скалярная величина,

$x_0$  ( $x_0 \geq 0$ ) — дополнительная переменная,

$c_0 - x_0 + cx = 0$  — уравнение связи дополнительной переменной,

$\min_{x \in R^n} (x_0), c_0 - x_0 + cx = 0, h + Hx \geq 0, x_0 \geq 0, x \geq 0$ , — задача, эквивалентная задаче Klee–Schrijver.

$x_{n+1} \geq 0, \dots, x_{2n} \geq 0$  — дополнительные переменные для приведения задачи Klee–Schrijver к каноническому виду путем добавления к правой части каждого уравнения соответственно.

**Шаг 0** (инициализация). Вычислить номер  $r$  ведущего столбца (номер ведущей строки  $q = 0$ ) по формуле

$$\Theta_r = \min_{i=1, \dots, n, c_i < 0} c_i = c_1 = -2^{n-1}.$$

Переписать уравнение связи дополнительной переменной  $x_0 \geq 0$ :

$$\begin{aligned} x_1 &= x_1^0 + e_{1_0}^0 \bullet (x_0 \ x_2 \ \dots \ x_n)^T, \\ x_1^0 &= 2^{-n+1} c_0, \quad e_{1_0}^0 \bullet = (-2^{-n+1} \ -2^{-1} \ \dots \ -2^{-n+2} \ -2^{-n+1}), \\ E_{1_0}^0 &= \begin{pmatrix} -2^{-n+1} & -2^{-1} & \dots & -2^{-n+2} & -2^{-n+1} \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}. \end{aligned}$$

Положить

$$x^0 = (2^{-n+1} c_0 \ 0 \ \dots \ 0)^T.$$

**Шаг 1** (расчет  $x^1$ ). Вычислить номер  $q$  ведущей строки по формуле

$$\theta_q = \min_{j=1, \dots, n, h_j + h_j \bullet x^0 < 0} (h_j + h_j \bullet x^0) = h_n + h_n \bullet x^0 = 5^n - 2c_0.$$

Вычислить элементы уравнения связи  $n$ -й строки:

$$\begin{aligned} h_n^1 + (h_{n1}^1 \ h_{n2}^1 \ \dots \ h_{nn}^1) (x_0 \ x_2 \ \dots \ x_n)^T &= x_{2n}, \\ h_n^1 = h_n + h_{n1} x_1^0 = 5^n - 2c_0, \quad (h_{n1}^1 \ h_{n2}^1 \ \dots \ h_{nn}^1) &= h_n E_{1_0}^0 \bullet = (2 \ 0 \ \dots \ 0 \ 1). \end{aligned}$$



Вычислить номер  $r$  ведущего столбца по формуле

$$\Theta_r = \max_{i=1, \dots, n, h_{ni}^1 > 0, i \neq 1} h_{ni}^1 = h_{nm}^1 = 1.$$

Переписать уравнение связи  $n$ -й строки:

$$\begin{aligned} x_n &= x_n^1 + e_{n_1}^1 \cdot (x_0 \ x_2 \ \dots \ x_{n-1} \ x_{2n})^T, \\ x_n^1 &= -5^n + 2c_0, \quad e_{n_1}^1 \cdot = (-2 \ 0 \ \dots \ 0 \ 1), \\ E_{n_1}^1 &= \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ -2 & 0 & \dots & 0 & 1 \end{pmatrix}. \end{aligned}$$

Пересчитать элементы уравнения связи:

$$\begin{aligned} x_1 &= x_1^1 + e_{1_1}^0 \cdot (x_0 \ x_2 \ \dots \ x_{n-1} \ x_{2n})^T, \\ x_1^1 &= -2^{-n+1}c_0 + 2^{-n+1}5^n, \quad e_{1_1}^0 \cdot = e_{1_0}^0 \cdot E_{n_1}^1 = (2^{-n+1} \ -2^{-1} \ \dots \ -2^{-n+2} \ -2^{-n+1}). \end{aligned}$$

Переписать уравнение связи переменной  $x_1 \geq 0$  ( $x_1^1 < 0$ ):

$$\begin{aligned} x_0 &= x_0^1 + e_{0_1}^1 \cdot (x_1 \ x_2 \ \dots \ x_{n-1} \ x_{2n})^T, \\ x_0^1 &= -5^n + c_0, \quad e_{0_1}^1 \cdot = (2^{n-1} \ 2^{n-2} \ \dots \ 2 \ 1), \\ E_{0_1}^1 &= \begin{pmatrix} 2^{n-1} & 2^{n-2} & \dots & 2 & 1 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}. \end{aligned}$$

Пересчитать элементы уравнения связи:

$$\begin{aligned} x_n &= x_n^2 + e_{n_2}^1 \cdot (x_1 \ x_2 \ \dots \ x_{n-1} \ x_{2n})^T, \\ x_n^2 &= 5^n, \quad e_{n_2}^1 \cdot = e_{n_1}^1 \cdot E_{0_1}^1 = (-2^n \ -2^{n-1} \ \dots \ -2^2 \ -1). \end{aligned}$$

Положить

$$x^* = x^1 = (0 \ 0 \ \dots \ 0 \ 5^n)^T \geq 0, \quad cx^* = -5^n.$$

**ПРИМЕР 4** (Свириденко [Свириденко, 2017]). Большие задачи ЛП, как правило, имеют неединственное решение. Различные методы решения таких задач (симплекс-метод, метод внутренних точек, метод квадратичной штрафной функции) приводят к различным решениям в случае неединственности. Симплекс-метод дает решение, которое принадлежит вершине многогранного множества. Методы внутренней точки сходятся к решению, в котором выполнено условие строгой дополняющей нежесткости. Метод внешней квадратичной штрафной функции дает возможность найти точное нормальное решение (иными словами, решение, обладающее минимальной евклидовой нормой). Отметим, что нахождение нормального решения тесно связано

с методом регуляризации [Поляк, 2006] и методом квадратичной штрафной функции [Фиакко, Мак-Кормик, 1972].

Подход к построению всех вершин  $\{x^{1*}, \dots, x^{s*}\}$  многогранного множества решений задачи ЛП прямым алгоритмом рассмотрен на примере задачи вида (1) с бесконечным множеством решений [Свириденко, 2017]:

$$c = (1 \quad 1 \quad 2 \quad 2 \quad 1),$$

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & -1 & 2 & 2 & 3 \end{pmatrix}, \quad h = \begin{pmatrix} 2 \\ 2 \end{pmatrix},$$

$$x^{1*} = \left(0 \quad \frac{2}{3} \quad \frac{4}{3} \quad 0 \quad 0\right)^T, \quad cx^{1*} = \frac{10}{3},$$

$$x^{2*} = \left(0 \quad \frac{2}{3} \quad 0 \quad \frac{4}{3} \quad 0\right)^T, \quad cx^{2*} = \frac{10}{3},$$

$$x^{3*} = \left(0 \quad \frac{6}{3} \quad 0 \quad 0 \quad \frac{4}{3}\right)^T, \quad cx^{3*} = \frac{10}{3}.$$

Теперь рассмотрим подход к расчету нормального решения  $x^*$  как к решению задачи проекции нуля на множество решений задачи Свириденко:

$$\{\alpha_1 x^{1*} + \alpha_2 x^{2*} + \alpha_3 x^{3*} \mid \alpha_1 + \alpha_2 + \alpha_3 = 1, \alpha_s \geq 0, s = 1, 2, 3\}.$$

Решение этой задачи существует и единственно [Малозёмов, 2017b], ее решение:

$$\alpha_1 = \frac{1}{2}, \quad \alpha_2 = \frac{1}{2}, \quad \alpha_3 = 0,$$

следовательно,

$$x^* = \frac{1}{2}x^{1*} + \frac{1}{2}x^{2*} + 0x^{3*} = \left(0 \quad \frac{2}{3} \quad \frac{2}{3} \quad \frac{2}{3} \quad 0\right)^T, \quad cx^* = \frac{10}{3}.$$

## Заключение

Прямой метод представляет собой последовательность шагов, на каждом из которых получают нули в нужных позициях очередного обрабатываемого столбца матрицы условий прямой задачи. При этом сохраняются нули, полученные ранее в предыдущих столбцах.

Лачинов и Поляков в [Лачинов, Поляков, 1999] отметили, что любой современный язык программирования имеет «вкус» и «запах» своих предшественников, так и прямой алгоритм неизбежно несет черты симплекс-метода с обратной матрицей. Более того, предложены два варианта алгоритма решения задачи, полученной на первом этапе, в сопряженной канонической форме. Первый основан на ее решении прямым алгоритмом в терминах симплекс-метода, а второй — на решении задачи, двойственной к ней, симплекс-методом. Доказано, что оба варианта эквивалентны: при одинаковых исходных данных генерируют одинаковую последовательность точек — базисное решение и текущее двойственное решение вектора оценок строк. Отсюда следует вывод: прямой алгоритм — это алгоритм типа симплекс-метода.

На каждой итерации каждого этапа прямой алгоритм генерирует последовательность точек, выходящих за пределы допустимой области в направлении возрастания целевой функции, до тех пор, пока не будет получено допустимое решение, причем гиперплоскость целевой функции, проходящая через эти точки, не пересекает допустимую область. Поэтому, в соответствии с терминологией Фиакко и Мак-Кормика, прямой алгоритм относится к одному из вариантов метода внешних точек [Фиакко, Мак-Кормик, 1972].

Симплекс-метод, метод внутренних точек и метод квадратичной штрафной функции приводят к различным решениям в случае их неединственности. Симплекс-метод дает решение, которое принадлежит вершине многогранного множества. Методы внутренней точки сходятся к решению, в котором выполнено условие строгой дополняющей нежесткости. Метод внешней квадратичной штрафной функции дает возможность найти решение, обладающее минимальной евклидовой нормой (точное нормальное решение). Решением задачи из примера 4 (подход к расчету нормального решения) покажем, что прямой метод дает возможность найти все оптимальные вершины  $(x^1, x^2, x^3)$  многогранного множества и, следовательно, приводит и к точному нормальному решению, и к решению, в котором выполнено условие дополняющей нежесткости. Решение  $(x^1)$ , принадлежащее вершине многогранного множества, достигается на второй итерации первого этапа:

$$\min(x_1), \quad x_2 = \frac{2}{3} + \frac{1}{3}x_1 + x_5, \quad x_3 = \frac{4}{3} - \frac{1}{3}x_1 - x_4 - x_5, \quad x_i \geq 0, \quad i = 1, \dots, 5,$$

$$x^1 = \left(0 \quad \frac{2}{3} \quad \frac{4}{3} \quad 0 \quad 0\right)^T, \quad cx^1 = \frac{10}{3}.$$

Для поиска остальных решений  $(x^2, x^3)$ , принадлежащих вершинам многогранного множества, решим задачи ЛП (оптимальное значение  $x_1$  заведомо равно нулю):

$$\max(x_4), \quad x_2 = \frac{2}{3} + x_5, \quad x_3 = \frac{4}{3} - x_4 - x_5, \quad x_i \geq 0, \quad i = 2, \dots, 5,$$

$$\max(x_5), \quad x_2 = \frac{2}{3} + x_5, \quad x_3 = \frac{4}{3} - x_4 - x_5, \quad x_i \geq 0, \quad i = 2, \dots, 5.$$

Их решения:

$$(x_2 \ x_3 \ x_4 \ x_5)^T = \left(\frac{2}{3} \ 0 \ \frac{4}{3} \ 0\right)^T, \quad (x_2 \ x_3 \ x_4 \ x_5)^T = \left(0 \ \frac{6}{3} \ 0 \ \frac{4}{3}\right)^T,$$

следовательно,

$$x^2 = \left(0 \ \frac{2}{3} \ 0 \ \frac{4}{3} \ 0\right)^T, \quad cx^2 = \frac{10}{3}, \quad x^3 = \left(0 \ \frac{6}{3} \ 0 \ 0 \ \frac{4}{3}\right)^T, \quad cx^3 = \frac{10}{3}.$$

## Благодарности

Автор выражает признательность рецензенту за внушительный вклад в написание работы.

## Список литературы (References)

- Агафонова И. В., Даугавет В. А. Вырожденность в задачах линейного программирования // Избранные лекции по экстремальным задачам. Часть 1 / под ред. проф. В. Н. Малозёмова. — СПб.: Изд-во ВВМ, 2017. — С. 37–42.
- Agafonova I. V., Daugavet V. A. Vyrozhdennost' v zadachah linejnogo programmirovaniya [Degeneracy in linear programming problems] // Selected lectures on extremal problems. Part 1 / ed. prof. V. N. Malozemova. — St. Petersburg: Izd-vo VVM, 2017. — P. 37–42 (in Russian).
- Беников А. И. Линейное программирование: учебное пособие. — Иркутск: Иркутский университет, 2005. — 147 с.
- Benikov A. I. Lineinoe programmirovanie: uchebnoe posobie [Linear programming: a tutorial]. — Irkutsk: Irkutsk University, 2005. — 147 p. (in Russian).
- Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. — М.: Мир, 1985.
- Gill Ph. E., Murray W., Wright M. H. Practical optimization. — System Optimization Laboratory Department of Operations Research Stanford University California, USA; Academic Press, 1981. (Russ. ed.: Gill F., Mjurrej U., Rajt M. Prakticheskaja optimizacija. — Moscow: Mir, 1985.)

- Голиков А. И., Евтушенко Ю. Г. Отыскание нормальных решений в задачах линейного программирования // Журнал вычислительной математики и математической физики. — 2000. — Т. 40, № 12. — С. 1766–1792.  
*Golikov A. I., Evtushenko Yu. G.* Otyskanie normal'nyh reshenij v zadachah linejnogo programmirovaniya [Finding normal solutions in linear programming problems] // Journal of computational mathematics and mathematical physics. — 2000. — Vol. 40, No. 12. — P. 1766–1792 (in Russian).
- Золотых Н. Ю., Кубарев В. К. Полностью целочисленный разреженный симплекс-метод // Вестник Нижегородского университета им. Н. И. Лобачевского. — 2012. — № 4 (1). — С. 232–237.  
*Zolotyh N. Yu., Kubarev V. K.* Polnost'yu celochislennyj razrezhenyj simpleks-metod [A fully integer sparse simplex method] // Bulletin of Nizhny Novgorod University N. I. Lobachevsky. — 2012. — No. 4 (1). — P. 232–237 (in Russian).
- Кузюрин Н. Н., Фомин С. А. Эффективные алгоритмы и сложность вычислений. — М.: МФТИ, 2007.  
*Kuzjurin N. N., Fomin S. A.* Jeffektivnye algoritmy i slozhnost' vychislenij [Efficient algorithms and computational complexity]. — Moscow: MFTI, 2007 (in Russian).
- Лачинов В. М., Поляков А. О. Информодинамика или путь к миру открытых систем // СПб.: Изд-во СПбГТУ, 1999.  
*Lachinov V. M., Polyakov A. O.* Informodinamika ili put' k miru otkrytyh sistem [Informdynamics or the path to the world of open systems] // St. Petersburg: SPbSTU Publishing house, 1999 (in Russian).
- Майстренко А. В., Светлаков А. А., Черепанов Р. О. Исследование свойств матрицы Гильберта и причин ее плохой обусловленности // Омский научный вестник. — 2011. — № 3 (103). — С. 265–269.  
*Majstrenko A. V., Svetlakov A. A., Cherepanov R. O.* Issledovanie svojstv matricy Gil'berta i prichin ee plohoj obuslovlennosti [Investigation of the properties of the Hilbert matrix and the reasons for its bad conditionality] // Omskij nauchnyj vestnik. — 2011. — No. 3 (103). — P. 265–269 (in Russian).
- Малков У. Х. Обзор путей повышения эффективности мультипликативного алгоритма симплекс-метода // Математические методы решения экономических задач: сборник. — М.: Наука, 1977. — № 7.  
*Malkov U. Kh.* Obzor putej povysheniya effektivnosti mul'tiplikativnogo algoritma simpleks-metoda [Review of ways to increase the efficiency of the multiplicative simplex method] // Mathematical methods for solving economic problems: collection. — Moscow: Nauka, 1977. — No. 7 (in Russian).
- Малозёмов В. Н. Модифицированный симплекс-метод // Избранные лекции по экстремальным задачам. Часть 1 / под ред. проф. В. Н. Малозёмова. — СПб.: Изд-во ВВМ, 2017а. — С. 15–24.  
*Malozyomov V. N.* Modificirovannyj simpleks-metod [Modified simplex method] // Selected lectures on extremal problems. Part 1 / ed. prof. V. N. Malozemov. — St. Petersburg: Izd-vo VVM, 2017. — P. 15–24 (in Russian).
- Малозёмов В. Н. О задаче проектирования нуля на многогранник // Избранные лекции по экстремальным задачам. Часть 1 / под ред. проф. В. Н. Малозёмова. — СПб.: Изд-во ВВМ, 2017б. — С. 182–187.  
*Malozyomov V. N.* O zadache proektirovaniya nulya na mnogogrannik (On the problem of designing a zero on a polyhedron) // Selected lectures on extremal problems. Part 1 / ed. prof. V. N. Malozemov. — St. Petersburg: Izd-vo VVM, 2017. — P. 182–187 (in Russian).
- Моисеев Н. Н., Иванюлов Ю. П., Столярова Е. М. Методы оптимизации. — М.: Наука, Главная редакция физико-математической литературы, 1978.  
*Moiseev N. N., Ivanilov Yu. P., Stolyarova E. M.* Metody optimizacii [Optimization method]. — Moscow: Nauka, Main edition of physical and mathematical literature, 1978 (in Russian).
- Поляк Б. Т. Метод Ньютона и его роль в оптимизации и вычислительной математике // Труды ИСА РАН. — 2006. — Т. 28. — С. 48–66.  
*Poljak B. T.* Metod N'jutona i ego rol' v optimizacii i vychislitel'noj matematike [Newton's method and its role in optimization and computational mathematics] // Proceedings of ISA RAS. — 2006. — Vol. 28. — P. 48–66 (in Russian).
- Свириденко А. Б. Априорная поправка в ньютоновских методах оптимизации // Компьютерные исследования и моделирование. — 2015. — Т. 7, № 4. — С. 835–863.  
*Sviridenko A. B.* Apriornaja popravka v n'jutonovskih metodah optimizacii [The correction to Newton's methods of optimization] // Computer Research and Modeling. — 2015. — Vol. 7, No. 4. — P. 835–863 (in Russian).
- Свириденко А. Б. О проектировании нуля на линейное многообразие, многогранник и вершину многогранника. Ньютоновские методы минимизации // Компьютерные исследования и моделирование. — 2019. — Т. 11, № 4. — С. 563–591.  
*Sviridenko A. B.* O proektirovanii nulya na linejnoe mnogoobrazie, mnogogrannik i vershinu mnogogrannika. N'jutonovskie metody minimizacii [Designing a zero on a linear manifold, a polyhedron, and a vertex of a polyhedron. Newton methods of minimization] // Computer research and modeling. — 2018. — Vol. 11, No. 4. — P. 563–591 (in Russian).

- Свириденко А. Б.* Прямые мультипликативные методы для разреженных матриц. Линейное программирование // Компьютерные исследования и моделирование. — 2017. — Т. 9, № 2. — С. 143–165.  
*Sviridenko A. B.* Prjamyje mul'tiplikativnye metody dlja razrezhennyh matric. Nesimmetrichnye linejnye sistemy [Direct multiplicative methods for sparse matrices. Linear Programming] // Computer research and modeling. — 2017. — Vol. 9, No. 2. — P. 143–165 (in Russian).
- Свириденко А. Б.* Прямые мультипликативные методы для разреженных матриц. Несимметричные линейные системы // Компьютерные исследования и моделирование. — 2016. — Т. 8, № 6. — С. 833–860.  
*Sviridenko A. B.* Prjamyje mul'tiplikativnye metody dlja razrezhennyh matric. Nesimmetrichnye linejnye sistemy [Direct multiplicative methods for sparse matrices. Unbalanced linear systems] // Computer research and modeling. — 2016. — Vol. 8, No. 6. — P. 833–860 (in Russian).
- Схрейвер А.* Теория линейного и целочисленного программирования. — М.: Мир, 1991.  
*Schrijver A.* Theory of linear and integer programming. — Department of Econometrics, Tilburg University and Centrum voor Wiskunde en Informatica, Amsterdam; A Wiley-Interscience Publication, John Wiley & sons, 1990. (Russ. ed.: *Shrejver A.* Teorija linejnogo i celochislennogo programmirovanija. — Mocsow: Mir, 1991.)
- Фиакко А., Мак-Кормик Г.* Нелинейное программирование. Методы последовательной безусловной минимизации. — М.: Мир, 1972.  
*Fiacco A. V., McCormick G. P.* Nonlinear programming: sequential unconstrained minimization techniques. — Research analysis corporation McLean, Virginia; John Wiley and cons, Inc., New York, London, Sydney, Toronto, 1968. (Russ. ed.: *Fiakko A., Mak-Kormik G.* Nelinejnoe programmirovanie. Metody posledovatel'noj bezuslovnoj minimizacii. — Mocsow: Mir, 1972.)
- Хакимова А. Б., Зеленков Г. А., Рзун И. Г.* Подход к увеличению эффективности мультипликативного алгоритма симплекс-метода // Труды ИСА РАН «Динамика неоднородных систем». — 2010. — Т. 53 (2), вып. 14. — С. 245–251.  
*Khakimova A. B., Zelenkov G. A., Rzun I. G.* Podhod k uvelicheniju jeffektivnosti mul'tiplikativnogo algoritma simpleks-metoda [Approach to increase the efficiency of the multiplicative algorithm of the simplex method] // The works of ISA Russian Academy of Sciences “Dynamics of heterogeneous systems”. — 2010. — Vol. 53 (2), Iss. 14. — P. 245–251 (in Russian).
- Шевченко В. Н., Золотых Н. Ю.* Линейное и целочисленное линейное программирование. — Нижний Новгород: Изд-во Нижегородского госуниверситета им. Н. И. Лобачевского, 2004. — 154 с.  
*Shevchenko V. N., Zolotyh N. Yu.* Linejnoe i celochislennoe linejnoe programmirovanie [Linear and integer linear programming]. — Nizhnij Novgorod: Publishing house of Nizhny Novgorod state University N. I. Lobachevsky, 2004. — 154 p. (in Russian).
- Шевченко В. Н., Золотых Н. Ю.* Линейное и целочисленное программирование: учебное пособие. — Нижний Новгород: Изд. Нижегородского государственного университета, 2002.  
*Shevchenko V. N., Zolotyh N. Yu.* Linejnoe i celochislennoe programmirovanie [Linear and integer programming: textbook. — Nizhny Novgorod: Nizhny Novgorod State University, 2002 (in Russian).
- Yan Z.* Simplex methods meeting the conjectured Hirsch // Semantic Scholar library. — [Electronic resource]. — <https://www.semanticscholar.org/paper/SIMPLEX-METHODS-MEETING-THE-CONJECTURED-HIRSCH-Zizong/efab4890e8bc0262d49e52e57151506d332b813e> (accessed: 16.03.2020).