COMPUTER RESEARCH AND MODELING
2024 VOL. 16 NO. 2 P. e249–e285
DOI: 10.20537/2076-7633-2024-16-2-249-285

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

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 nondegenerate 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, the very structure of which allows: 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 to 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 similar to the simplex method. It is also shown that the comparison of numerical schemes leads to the conclusion that the direct algorithm allows one 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. e249–e285 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2024 Т. 16 № 2 С. e249–e285
DOI: 10.20537/2076-7633-2024-16-2-249-285

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

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

Introduction

The LP problem in canonical form is considered:

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

where H — $(m\times n)$ is the matrix in which $m < n$, $c^T \in R^n$, $h \in R^m$, x is a vector of direct variables, and the elements of vectors c^T, h and matrix H are arbitrary real numbers.

The final algorithm for solving problem (1), based on the simplex method with an inverse matrix (modified simplex method), consists of two successive stages at which LP problems are solved using the simplex method: an auxiliary problem at the first stage, and some problem equivalent to the original one at the second [Агафонова, Даугавет, 2017]. The algorithm does not require any preliminary calculations or imposition of any additional conditions on c, h, H, including the condition of nondegeneracy of the problem, which ensures the final convergence of the algorithm. Final convergence is guaranteed by applying Bland's rule to avoid loops or any other method of eliminating loops. The algorithm is based on searching for a solution among the vertices of a polyhedron of feasible solutions. The search principle is as follows. First, one of the vertices is found, and then the vertex closest to it is searched for, at which the value of the objective function decreases. As soon as the transition from one vertex to another with a smaller value of the objective function becomes impossible, it is concluded that a solution has been found. The main disadvantage of the algorithm is its complexity. Basically, this is the need to transfer information (basis, plan and optimal value of the objective function [Агафонова, Даугавет, 2017]) to the next stage of the algorithm and use one of the methods to eliminate looping to guarantee final convergence.

In this work, which is a continuation of previous research [Свириденко, 2015; Свириденко, 2016; Свириденко, 2017; Свириденко, 2019], the basis for eliminating the shortcomings of the simplex method with an inverse matrix is the refusal to search for a solution among the vertices of the polyhedron of feasible solutions and the transition to a direct search. To simplify, we will assume that $c \geqslant 0$, otherwise the following auxiliary quantities will be involved in the description:

c_0 $\left(-c_0 \leqslant cx^*\right)$ is an arbitrarily large scalar quantity,

x_0 $(x_0 \geqslant 0)$ is an additional variable,

$c_0 - x_0 + cx = 0$ is the constraint equation of an additional variable,

$\min\limits_{x\in R^n}(x_0)$, $c_0 - x_0 + cx = 0$, $h + Hx = 0$, $x_0 \geqslant 0$, $x \geqslant 0$, is a problem equivalent to (1).

Technically, this is done as shown in the description of the computational scheme of the first stage of the direct algorithm in terms of the simplex method in Section 3.2 and using examples of solving problems with guaranteed behavior of the simplex method in Section 4. We will also assume that problem (1), perhaps degenerate, has a unique solution, or any admissible point is optimal, and the latter occurs if and only if there is a linear combination of c rows of the matrix H, i. e., there is an m-dimensional row vector w such that $c = wH$. It follows that the objective function of problem (1) is a constant ($cx = wHx = -wh$), and it is impossible to optimize the constant. Strictly speaking, such a problem is not an LP problem; it is a classical problem of a conditional extremum from the course of mathematical analysis [Беников, 2005].

1. Rationale for the algorithm

Let's denote:

k is the iteration number,

$h_{k\bullet} = \begin{pmatrix} h_{k1} & h_{k2} & \cdots & h_{kn}\end{pmatrix}$ is the kth row of the matrix H,

$h_k^k + \begin{pmatrix} h_{kk}^k & h_{kk+1}^k & \cdots & h_{kn}^k\end{pmatrix}\begin{pmatrix} x_k & x_{k+1} & \cdots & x_n\end{pmatrix}^T = 0$ is the constraint equation of the kth row,

$x^k = \begin{pmatrix} x_1^k & x_2^k & \cdots & x_k^k & 0 & \cdots & 0 \end{pmatrix}^T$ is the n-dimensional vector, the solution of a system of linear equations:

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

$e_{1_k\bullet}^k = \begin{pmatrix} e_{1_k k+1}^k & e_{1_k k+2}^k & \cdots & e_{1_k n}^k \end{pmatrix}$ is the the main line of the multiplier:

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

Here $E_{1_k}^k$ $(n-k)\times(n-k-1)$ is the matrix, the first row of which is arbitrary, and the remaining rows are the rows of the $(n-k-1)\times(n-k-1)$ identity matrix. $E_{r_k}^k$ (r_k is the number of the leading line at the iteration k) coincides with the multiplier in the simplex method algorithm up to the transposition sign [Хакимова, Зеленков, Рзун, 2010; Свириденко, 2015; Свириденко, 2017], with the exception of removing the leading column, all elements of which are equal to zero. The presented form of the multiplier $E_{1_k}^k$ is an artifact and is caused by the assumption $r = k$ that is made below in step 3 of the first stage of the algorithm.

With the introduced notation, the principle of searching for a solution using a two-stage algorithm is as follows. The polyhedron of feasible solutions is not specified initially, as in the simplex method, but is formed by constructing the following problem at each iteration $k = 1, \ldots, m$:

$$\min_{x\in R^n}\left(c_{k+1}^k x_{k+1} + c_{k+2}^k x_{k+2} + \cdots + c_n^k x_n\right),$$
$$x_i = x_i^k + e_{1_k k+1}^i x_{k+1} + e_{1_k k+2}^i x_{k+2} + \cdots + e_{1_k n}^i x_n, \quad i = 1, \ldots, k, \tag{2}$$
$$x_i \geqslant 0, \quad i = 1, \ldots, n,$$

in canonical form, equivalent to the problem:

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

(2) is based on a multiplicative version of the Gaussian elimination method [Свириденко, 2016], the very structure of which contains the following capabilities: identification of incompatibility and linear dependence of constraints; identification of variables whose optimal values are known to be zero; actual elimination of direct variables and reduction of the dimension of the space in which the solution to the problem (1) is determined. The possibilities listed above determine the construction of (2) at the first stage of the algorithm, subject to the optional fulfillment of the following conditions:

$$x_i = x_i^k \geqslant 0, \quad i = 1, \ldots, k.$$

The optionality of fulfilling all restrictions for the current solution provides greater flexibility of the algorithm in the case of “outs” from the admissible region and using the found solution as the initial one for the same problem with changed conditions.

The initial polyhedron is defined by the conditions for nonnegativity of variables, on which the minimum value of the objective function is equal to zero. Then the first constraint equation is added to its system of conditions and the optimum on the formed polyhedron is calculated by increasing (due to compression of the region of feasible solutions) the value of the objective function by some nonnegative constant. Calculations continue until m the last constraint is added to the system of conditions at an iteration. The first stage of the algorithm is described; technically, at each iteration it is done like this (a detailed description of all operations performed in constructing (2) is given in Section 2.1).

1.1. The first stage of the algorithm

Step 1. To construct the constraint equations:

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

of the kth line of the system of constraints:

$$h_k + h_{k\bullet} x = 0.$$

If $k = 1$, then set

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

If $k \geqslant 2$, then calculate using the formulas:

$$h_k^k = -h_k + \sum_{i=1}^{k-1} h_{ki} x_i^i, \quad \begin{pmatrix} h_{kk}^k & h_{kk+1}^k & \cdots & h_{kn}^k \end{pmatrix} = h_{k\bullet} \prod_{i=1}^{k-1} E_{1_{k-1}}^i.$$

REMARK. The formulas of step 1 are the result of the substitution:

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

into the kth line of the system of constraints.

Without loss of generality, for simplicity we assume:

$$h_k^k \leqslant 0,$$

otherwise it is enough to set:

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

Step 2. Analyze the constraint equation (identification of inconsistency and linear dependence of constraints; identification of variables whose optimal values are known to be zero).

Step 3. To calculate the number r of the leading column according to the rule:

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

Without loss of generality, we will set:

$$r = k.$$

Rewrite the constraint equation:

$$x_r = x_k = x_k^k + e_{1_k\bullet}^k \begin{pmatrix} x_{k+1} & x_{k+2} & \cdots & x_n \end{pmatrix}^T.$$

If $k = 1$, then

$$c^k = \begin{pmatrix} c_{k+1}^k & c_{k+2}^k & \cdots & c_n^k \end{pmatrix} = cE_{1_k}^k,$$

$$x^k = \begin{pmatrix} x_1^k & 0 & \cdots & 0 \end{pmatrix}^T.$$

If $k \geqslant 2$, then calculate using the formulas:

$$c^k = \begin{pmatrix} c_{k+1}^k & c_{k+2}^k & \cdots & c_n^k \end{pmatrix} = c^{k-1} E_{1_k}^k,$$

$$e_{1_k\bullet}^i = e_{1_{k-1}\bullet}^i E_{1_k}^k, \quad x_i^k = x_i^i + x_k^k e_{1_{k-1}k}^i, \quad i = 1, \ldots, k-1,$$

$$x^k = \begin{pmatrix} x_1^k & x_2^k & \cdots & x_k^k & 0 & \cdots & 0 \end{pmatrix}^T.$$

REMARK. The formulas of step 3 are the result of substitution x_k into the conditions of problem (2).

The rule for selecting the leading column guarantees the positivity c^k at each iteration $k = 1, \ldots, m$, therefore, if $x^m \geqslant 0$, then the algorithm becomes one-stage (see Example 1 in § 4 for looping the simplex method); otherwise, we proceed to the description of the second stage.

1.2. The second stage of the algorithm

To simplify the description, let us denote by ν the number of iterations and set for $\nu = 0$:

$$x = x_\nu = \begin{pmatrix} x_{\nu_1} & x_{\nu_2} & \cdots & x_{\nu_n} \end{pmatrix}^T,$$
$$x^m = x^\nu_\nu = \begin{pmatrix} x^\nu_{\nu_1} & x^\nu_{\nu_2} & \cdots & x^\nu_{\nu_m} & 0 & \cdots & 0 \end{pmatrix}^T,$$
$$c^m = c^\nu_\nu = \begin{pmatrix} c^\nu_{\nu_{m+1}} & c^\nu_{\nu_{m+2}} & \cdots & c^\nu_{\nu_n} \end{pmatrix},$$
$$e^i_{1_m\bullet} = e^\nu_{\nu_i\bullet} = \begin{pmatrix} e^\nu_{\nu_i m+1} & e^\nu_{\nu_i m+2} & \cdots & e^\nu_{\nu_i n} \end{pmatrix}, \quad i = 1, \ldots, m.$$

With the introduced notation, we rewrite (2) as

$$\begin{gathered} \min_{x_\nu \in R^n} \left(c^\nu_{\nu_{m+1}} x_{\nu_{m+1}} + c^\nu_{\nu_{m+2}} x_{\nu_{m+2}} + \cdots + c^\nu_{\nu_n} x_{\nu_n} \right), \\ x_{\nu_i} = x^\nu_{\nu_i} + e^\nu_{\nu_i m+1} x_{\nu_{m+1}} + e^\nu_{\nu_i m+2} x_{\nu_{m+2}} + \cdots + e^\nu_{\nu_i n} x_{\nu_n}, \quad i = 1, \ldots, m, \\ x_{\nu_i} \geqslant 0, \quad i = 1, \ldots, n. \end{gathered} \tag{3}$$

From a mathematical point of view, (3) is equivalent to the problem in conjugate canonical form:

$$\begin{gathered} \min_{x_\nu \in R^{n-m}} \left(c^\nu_{\nu_{m+1}} x_{\nu_{m+1}} + c^\nu_{\nu_{m+2}} x_{\nu_{m+2}} + \cdots + c^\nu_{\nu_n} x_{\nu_n} \right), \\ x^\nu_{\nu_i} + e^\nu_{\nu_i m+1} x_{\nu_{m+1}} + e^\nu_{\nu_i m+2} x_{\nu_{m+2}} + \cdots + e^\nu_{\nu_i n} x_{\nu_n} \geqslant 0, \quad i = 1, \ldots, m, \\ x_{\nu_i} \geqslant 0, \quad i = m+1, \ldots, n. \end{gathered} \tag{4}$$

REMARK. x_{ν_i} in (3) can be considered as additional, their values are equal to $x^\nu_{\nu_i}$. Section 2.3 shows that conditions (3) are redefined and that, in fact, (3) and (4) are the same problem. Moreover, an admissible solution to the problem dual to (4) is known — a zero m-dimensional vector. The algorithm of solving this problem is discussed in Section 3.

At each iteration of the second stage, a constraint is introduced into the calculation according to the formula:

$$x_{\nu_i} = x^\nu_{\nu_i} + e^\nu_{\nu_i m+1} x_{\nu_{m+1}} + e^\nu_{\nu_i m+2} x_{\nu_{m+2}} + \cdots + e^\nu_{\nu_i n} x_{\nu_n}, \quad x^\nu_{\nu_i} < 0,$$

by increasing (due to compression of the region of feasible solutions) the value of the objective function by some positive constant. Calculations continue until the minimum value of the objective function on the polyhedron of the final configuration is found. Technically, at each iteration this is done as follows (a detailed description of all operations performed during the solution (3) is given in Section 2.2).

Step 1. Calculate the number ν_q of the leading constraint equation using the formula:

$$x_{\nu_q} = x^\nu_{\nu_q} + e^\nu_{\nu_q m+1} x_{\nu_{m+1}} + e^\nu_{\nu_q m+2} x_{\nu_{m+2}} + \cdots + e^\nu_{\nu_q n} x_{\nu_n}, \quad x^\nu_{\nu_q} < 0.$$

Calculate the number ν_r of the leading column according to the rule:

$$\Theta_{\nu_r} = \min_i \frac{c^\nu_{\nu_i}}{e^\nu_{\nu_q i}}, \quad e^\nu_{\nu_q i} > 0, \quad i = m+1, \ldots, n.$$

Set: $\nu = \nu + 1$.

Step 2. Calculate:

$$x^\nu_{\nu_r} = -\frac{x^{\nu-1}_{\nu-1_q}}{e^{\nu-1}_{\nu-1_q r}}, \quad x^\nu_{\nu_i} = x^{\nu-1}_{\nu-1_i} + e^{\nu-1}_{\nu-1_i r} x^\nu_{\nu_r}, \quad i \neq r.$$

Set:

$$x_\nu^\nu = \begin{pmatrix} x_{\nu_1}^\nu & x_{\nu_2}^\nu & \cdots & x_{\nu_m}^\nu & 0 & \cdots & 0 \end{pmatrix}^T.$$

If $x_\nu^\nu \geqslant 0$, then stop.

Step 3. Rewrite the constraint equation:

$$x_{\nu_r} = x_{\nu_r}^\nu + e_{\nu_r m+1}^\nu x_{\nu_{m+1}} + e_{\nu_r m+2}^\nu x_{\nu_{m+2}} + \cdots + e_{\nu_r n}^\nu x_{\nu_n}.$$

Calculate according to the formulas:

$$e_{\nu_i \bullet}^\nu = e_{\nu-1_i \bullet}^{\nu-1} E_{\nu_r}^\nu, \quad i \neq q,$$

$$c_\nu^\nu = \begin{pmatrix} c_{\nu_{m+1}}^\nu & c_{\nu_{m+2}}^\nu & \cdots & c_{\nu_n}^\nu \end{pmatrix} = c_{\nu-1}^{\nu-1} E_{\nu_r}^\nu$$

and go to step 1.

In the case of nonuniqueness of the solution, the direct algorithm, like the simplex method, provides a solution that belongs to the vertex of the polyhedral set. Without loss of generality, we will set

$$x_\nu^\nu \geqslant 0, \quad c_{\nu_i}^\nu = 0 \ (i = m+1, \ldots, m_0), \quad c_{\nu_i}^\nu > 0 \ (i = m_0 + 1, \ldots, n).$$

This means that the optimal values of the variables

$$x_{\nu_i} \ (i = m_0 + 1, \ldots, n)$$

are obviously equal to zero, so they can be excluded from consideration. Therefore, any solution of the system

$$x_{\nu_i} = x_{\nu_i}^\nu + e_{\nu_i m+1}^\nu x_{\nu_{m+1}} + e_{\nu_i m+2}^\nu x_{\nu_{m+2}} + \cdots + e_{\nu_i n}^\nu x_{\nu_{m_0}} \ (i = 1, \ldots, m), \quad x_{\nu_i} \geqslant 0 \ (i = 1, \ldots, m_0)$$

is optimal. Thus, the nonuniqueness of a solution means uncertainty and the possibility of choosing the best one according to one or another additional criterion, based on unformalized (heuristic) ideas about the object. It is known that the interior point method and the quadratic penalty function method lead to different solutions if they are not unique. Interior point methods converge to a solution in which the strict complementary nonrigidity condition is satisfied . The external quadratic penalty function method makes it possible to find a solution with a minimum Euclidean norm. The paper shows that the direct algorithm makes it possible to find all the optimal vertices of a polyhedral set and, therefore, leads to both an exact normal solution and a solution in which the complementary slackness condition is satisfied (see Example 4 in § 4 and the conclusion).

1.3. Notes on this section

The direct method is a sequence of steps, at each of which zeros are obtained in the required positions of the next processed column of the matrix of conditions of the direct problem. In this case, the zeros obtained previously in preceding columns are preserved. This definition fully corresponds to the construction (3) at the first stage and to the recalculation of the main lines of the multipliers at the second. The leading column selection rule guarantees nonnegativity c_ν^ν at each iteration ν, and the uniqueness of solution (1), equivalent to (3), is their positivity. The exclusion of "extra" restrictions and variables, the optimal values of which are obviously equal to zero, guarantees the nondegeneracy of (3), and the strict increase of the objective function from iteration to iteration guarantees the finiteness of the number of iterations. Thus, there is no need to pass the information to the second stage and use one of the loop removal techniques to guarantee eventual convergence.

Problems (2), generated during the formation of a polyhedron of feasible solutions, determine options for constructing a direct algorithm. For example, for a one-stage option, solving (2) at each iteration is sufficient if $x^k \ngeq 0$. In other words, by combining stages at each iteration k. Single-stage algorithms are not considered in this work, because they exclude the possibility of constructing a two-stage version of the strongly polynomial simplex method with an inverse matrix.

At each iteration of each stage, the algorithm generates a sequence of points that extend beyond the feasible region in the direction of increasing the objective function, until a feasible solution is obtained, and the hyperplane of the objective function passing through these points does not intersect the feasible region. Therefore, in accordance with the terminology of Fiacco and McCormick, it can be called one of the variants of the external point method [Фиакко, Мак-Кормик, 1972].

Section 2.1 describes the computational scheme for constructing (2). During the construction process, the algorithm generates a sequence of multipliers, the main rows of which form the constraint matrix (2). Minimizing their filling is the main task when constructing the algorithm. The possibility of such minimization without loss of accuracy of the results is inherent in the very structure of direct methods, therefore the computational scheme of the first stage can be considered as an implementation of an algorithm for reducing the dimension of the problem. Section 2.2 presents a computational scheme for solving (3) in strongly polynomial time, i.e., the number of arithmetic operations (homogeneous complexity) is limited by a polynomial on the dimension of the problem [Кузюрин, Фомин, 2007]. To prove it, it is enough to show that the iterations do not exceed $\frac{m}{2}$. Section 2.3 presents forms of notation (3): canonical and conjugate to canonical. Section 2.4 gives a geometric interpretation of (3), as well as a geometric representation of the iterations of the simplex method and the direct algorithm. Section 3 discusses the relationship between the direct algorithm and the modified simplex method and the construction of an upper estimate for the number of iterations. Section 3.1 provides a computational scheme for the second stage of the modified simplex method, associated with the identification of a basic system of constraints, regardless of the method of its construction. In Sections 3.2, 3.3, the results obtained are used to construct computational schemes for the stages of the direct algorithm in terms of the simplex method. Section 3.4 compares the direct algorithm for solving (4) with the simplex method for solving the problem dual to (4). It is shown that the algorithms 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. This means that the direct algorithm is a variant of the simplex method. It is also shown there that a comparison of computational schemes leads to the conclusion that the direct algorithm makes it possible to reduce, according to the cubic law, the number of arithmetic operations required to solve the problem dual to (4) in comparison with the simplex method. In Section 3.5, the results obtained are used to construct an upper bound for the iterations of the simplex method for solving the problem dual to (4). It is shown that the number of iterations does not exceed $\frac{m}{2}$. This means that the simplex method, which allows solving problems of the form (1) in strongly polynomial time, can be constructed from two successive stages at which the following LP problems are solved: constructing (3) by a direct algorithm at the first stage, and solving the problem dual to (3) in the conjugate canonical form, using the simplex method, at the second stage.

2. Preliminary information, recording forms and geometric interpretation

Below is a detailed description of all operations performed in constructing (3). If during the calculations the elements become smaller in absolute value, the so-called critical value ε_0, then it is proposed to equate them to zero.

2.1. The first stage of the algorithm

Let

$\mathbf{x}^*$ denote the n-dimensional vector, the solution to the problem (1).

Step 0 (initialization). Set:

$$k = 1,$$
$$c^{k-1} = c = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}.$$

Calculate the number q of the leading line:

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

Swap the elements of the qth and kth rows of the matrix H and vector h. Renumber and remember the order of the lines.

Calculate the elements h_k^k, h_{ki}^k of the constraint equation of the kth row:

$$h_k^k + \begin{pmatrix} h_{kk}^k & h_{kk+1}^k & \cdots & h_{kn}^k \end{pmatrix} \begin{pmatrix} x_k & x_{k+1} & \cdots & x_n \end{pmatrix}^T = 0,$$
$$h_k^k = h_k, \quad \begin{pmatrix} h_{kk}^k & h_{kk+1}^k & \cdots & h_{kn}^k \end{pmatrix} = \begin{pmatrix} h_{kk} & h_{kk+1} & \cdots & h_{kn} \end{pmatrix}.$$

Here and in what follows, we will set $h_k^k \leqslant 0$, otherwise it is enough to put:

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

Calculate the number r of the leading column:

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

Swap the elements of the rth and kth columns of the matrix H and the row vector c^0. Renumber and remember the order of the unknowns.

Rewrite the constraint equation:

$$x_k = x_k^k + e_{1_k \bullet}^k \begin{pmatrix} x_{k+1} & x_{k+2} & \cdots & x_n \end{pmatrix}^T,$$
$$x_k^k = -\frac{h_k^k}{h_{kk}^k}, \quad e_{1_k \bullet}^k = -\frac{\begin{pmatrix} h_{kk+1}^k & h_{kk+2}^k & \cdots & h_{kn}^k \end{pmatrix}}{h_{kk}^k}.$$

Calculate:

$$c^k = \begin{pmatrix} c_{k+1}^k & c_{k+2}^k & \cdots & c_n^k \end{pmatrix} = c^{k-1} E_{1_k}^k.$$

Set:

$$x^k = \begin{pmatrix} x_1^k & 0 & \cdots & 0 \end{pmatrix}^T.$$

Go to step 6.

Step 1 (calculation h_k^k, h_{ki}^k). Calculate the number q of the leading line using the formula:

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

Swap the elements of the qth and kth rows of the matrix H and vector h. Renumber and remember the order of the lines.

Calculate:

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

Remark. For manual counting, the leading line selection strategy can be based on simplifying the calculations.

Step 2 $\left(\text{analysis } h_k^k,\ h_{ki}^k\right)$. Without loss of generality, in analyzing the elements of the constraint equation, the identification of the following situations can be excluded from further consideration:

- $h_k^k < 0$, $h_{ki}^k \leqslant 0$, $i = k, \ldots, n$, — inconsistency of the constraints of problem (1);
- $h_k^k = 0$, $h_{ki}^k = 0$, $i = k, \ldots, n$, — excluding the kth row from the constraints of problem (1) and, as a consequence, reducing the number of multipliers;
- $h_k^k = 0$, $h_{ki}^k \leqslant 0$ (or $h_{ki}^k \geqslant 0$), $i = k, \ldots, n$, — reducing the dimension of problem (1) and, as a consequence, reducing the number of nonzero elements of the main lines of the multipliers obtained in the previous steps.

Step 3 $\left(\text{calculation } x_k^k,\ e_{1_k\bullet}^k\right)$. Calculate the number r of the leading column:

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

Swap the elements of the rth and kth columns of the matrix H and the row vector c^0, renumber and remember the order of the unknowns.

Rewrite the constraint equation:

$$x_k = x_k^k + e_{1_k\bullet}^k \begin{pmatrix} x_{k+1} & x_{k+2} & \cdots & x_n \end{pmatrix}^T,$$

$$x_k^k = -\frac{h_k^k}{h_{kk}^k}, \quad e_{1_k\bullet}^k = -\frac{\begin{pmatrix} h_{kk+1}^k & h_{kk+2}^k & \cdots & h_{kn}^k \end{pmatrix}}{h_{kk}^k}.$$

Step 4 (calculation of c^k, x^k). Calculate:

$$c^k = \begin{pmatrix} c_{k+1}^k & c_{k+2}^k & \cdots & c_n^k \end{pmatrix} = c^{k-1} E_{1_k}^k, \quad x_i^k = x_i^i + x_k^k e_{1_{k-1}k}^i, \quad i = 1, \ldots, k-1.$$

Set:

$$x^k = \begin{pmatrix} x_1^k & x_2^k & \cdots & x_k^k & 0 & \cdots & 0 \end{pmatrix}^T.$$

Step 5 $\left(\text{recalculation } e_{1_{k-1}}^i\right)$. To calculate:

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

Without loss of generality, when analyzing elements x_i^k, $e_{1_k\bullet}^i$ we can exclude from further consideration the identification of the following situations:

- $x_i^k < 0$, $e^i_{1_k\bullet} \leqslant 0$ — inconsistency of the constraints of problem (1);
- $x_i^k = 0$, $e^i_{1_k\bullet} \leqslant 0$ or $e^i_{1_k\bullet} \geqslant 0$ — reducing the dimension of problem (1) and, as a consequence, the number of nonzero elements of the main lines of the multipliers obtained in the previous steps.

Step 6 (recalculation of k). Set: $k = k + 1$. If $k \neq m$, then go to step 1.

Step 7 (calculation of x^*). If $x^m \geqslant 0$, then calculate $x^m \Rightarrow x^*$ and stop; otherwise, go to step 1 of the second stage of solving the problem:

$$\begin{gathered}\min_{x\in R^n}\left(c^m_{m+1}x_{m+1} + c^m_{m+2}x_{m+2} + \cdots + c^m_n x_n\right),\\ x_i = x_i^m + e^i_{1_m m+1}x_{m+1} + e^i_{1_m m+2}x_{m+2} + \cdots + e^i_{1_m n}x_n \geqslant 0, \quad i = 1, \ldots, m,\\ x_i \geqslant 0, \quad i = m+1, \ldots, n.\end{gathered} \tag{5}$$

Below is a detailed description of all operations performed during the solution (5). If during the calculations the elements become smaller in absolute value, the so-called critical value ε_0, then it is proposed to equate them to zero.

2.2. The second stage of the algorithm

Step 0 (initialization). Set:

$$\nu = 0$$

and rewrite (5) as:

$$\begin{gathered}\min_{x_\nu\in R^n}\left(c^\nu_{\nu_{m+1}}x_{\nu_{m+1}} + c^\nu_{\nu_{m+2}}x_{\nu_{m+2}} + \cdots + c^\nu_{\nu_n}x_{\nu_n}\right),\\ x_{\nu_i} = x^\nu_{\nu_i} + e^\nu_{\nu_i m+1}x_{\nu_{m+1}} + e^\nu_{\nu_i m+2}x_{\nu_{m+2}} + \cdots + e^\nu_{\nu_i n}x_{\nu_n} \geqslant 0, \quad i = 1, \ldots, m,\\ x_{\nu_i} \geqslant 0, \quad i = m+1, \ldots, n.\end{gathered} \tag{6}$$

Step 1 (calculation of ν_q, ν_r). Calculate the number ν_q of the leading constraint equation using the formula:

$$\theta_{\nu_q} = \min_i x^\nu_{\nu_i}, \quad x^\nu_{\nu_i} < 0, \quad i = 1, \ldots, m.$$

Calculate the number ν_r of the leading column using the formula:

$$\Theta_{\nu_r} = \min_i \frac{c^\nu_{\nu_i}}{e^\nu_{\nu_q i}}, \quad e^\nu_{\nu_q i} > 0, \quad i = m+1, \ldots, n.$$

Step 2 (recalculating ν). Set $\nu = \nu + 1$.

Step 3 (calculating x^ν_ν). Calculate:

$$x^\nu_{\nu_r} = -\frac{x^{\nu-1}_{\nu-1_q}}{e^{\nu-1}_{\nu-1_q r}}, \quad x^\nu_{\nu_i} = x^{\nu-1}_{\nu-1_i} + e^{\nu-1}_{\nu-1_i r}x^\nu_{\nu_r}, \quad i \neq r.$$

Set:

$$x^\nu_\nu = \begin{pmatrix} x^\nu_{\nu_1} & x^\nu_{\nu_2} & \cdots & x^\nu_{\nu_m} & 0 & \cdots & 0\end{pmatrix}^T.$$

If $x^\nu_\nu \geqslant 0$, then calculate $x^\nu_\nu \Rightarrow x^*$ and stop.

Step 4 $\left(\text{calculation of } e^{\nu}_{\nu_i\bullet},\ c^{\nu}_{\nu}\right)$. Rewrite the constraint equation of the variable $x_{\nu_q} \geqslant 0$:

$$x_{\nu_r} = x^{\nu}_{\nu_r} + e^{\nu}_{\nu_r m+1} x_{\nu_{m+1}} + e^{\nu}_{\nu_r m+2} x_{\nu_{m+2}} + \dots + e^{\nu}_{\nu_r n} x_{\nu_n},$$

$$x^{\nu}_{\nu_r} = -\frac{x^{\nu-1}_{\nu-1_q}}{e^{\nu-1}_{\nu-1_q r}}, \quad e^{\nu}_{\nu_r r} = \frac{1}{e^{\nu-1}_{\nu-1_q r}}, \quad e^{\nu}_{\nu_r i} = -\frac{e^{\nu-1}_{\nu-1_q i}}{e^{\nu-1}_{\nu-1_q r}}, \quad i \neq r,$$

$$E^{\nu}_{\nu_r} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ e^{\nu}_{\nu_r m+1} & \cdots & e^{\nu}_{\nu_r r} & \cdots & e^{\nu}_{\nu_r n} \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}.$$

Calculate:

$$e^{\nu}_{\nu_i\bullet} = e^{\nu-1}_{\nu-1_i\bullet} E^{\nu}_{\nu_r}, \quad i \neq q,$$

$$c^{\nu}_{\nu} = \begin{pmatrix} c^{\nu}_{\nu_{m+1}} & c^{\nu}_{\nu_{m+2}} & \cdots & c^{\nu}_{\nu_n} \end{pmatrix} = c^{\nu-1}_{\nu-1} E^{\nu}_{\nu_r}$$

and go to step 1. Without loss of generality, we can exclude the analysis of elements $x^{\nu}_{\nu_i}$ to $e^{\nu}_{\nu_i\bullet}$ from further consideration to identify the following situations:

- $x^{\nu}_{\nu_i} < 0,\ e^{\nu}_{\nu_i\bullet} \leqslant 0$ — inconsistency of the constraints of problem (6);
- $x^{\nu}_{\nu_i} = 0,\ e^{\nu}_{\nu_i\bullet} \leqslant 0$ or $e^{\nu}_{\nu_i\bullet} \geqslant 0$ — reducing the dimension of the problem (6).

REMARK. $E^{\nu}_{\nu_r}$ $(n-m)\times(n-m)$ is a matrix, the ν_rth row of which is arbitrary, and the rest are rows of the $(n-m)\times(n-m)$ unit matrix. $E^{\nu}_{\nu_r}$ coincides with the multiplier in the simplex method algorithm up to the transposition sign [Хакимова, Зеленков, Рзун, 2010].

2.3. Forms for recording the problem solved at the second stage of the algorithm

Denote:

$\upsilon = n - m$,

$0_{m\upsilon}$ is the $(m \times \upsilon)$ matrix with zero elements,

I_s is the $(s \times s)$ identity matrix,

0_s is the zero s-dimensional vector,

$c' = \begin{pmatrix} c^{\nu}_{\nu_{m+1}} & c^{\nu}_{\nu_{m+2}} & \cdots & c^{\nu}_{\nu_n} \end{pmatrix}$ is the υ-dimensional row vector,

$c'' = \begin{pmatrix} 0^T_m & c' \end{pmatrix}$ is the n-dimensional row vector in block form,

$b = \begin{pmatrix} x^{\nu}_{\nu_1} & x^{\nu}_{\nu_2} & \cdots & x^{\nu}_{\nu_m} \end{pmatrix}^T$ is the m-dimensional vector,

$\overline{z} = \begin{pmatrix} b^T & 0^T_{\upsilon} \end{pmatrix}^T$ is the n-dimensional vector in block form,

$z = \begin{pmatrix} x_{\nu_1} & x_{\nu_2} & \cdots & x_{\nu_n} \end{pmatrix}^T$ is the n-dimensional vector of variables,

$y = \begin{pmatrix} z_{m+1} & z_{m+2} & \cdots & z_n \end{pmatrix}^T$ is the υ-dimensional vector of variables,

$N = \begin{pmatrix} e^{\nu}_{\nu_1 m+1} & \cdots & e^{\nu}_{\nu_1 n} \\ \vdots & \ddots & \vdots \\ e^{\nu}_{\nu_m m+1} & \cdots & e^{\nu}_{\nu_m n} \end{pmatrix}$ is the $(m \times \upsilon)$ matrix of rank m,

$A = \begin{pmatrix} I_m & -N \end{pmatrix}$ is the $(m \times n)$ matrix of rank m in block form,

$K = \begin{pmatrix} -N^T & -I_{\upsilon} \end{pmatrix}$ is the $(\upsilon \times n)$ matrix in block form.

With the introduced notation, problem (6) takes the form

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

where $z = \overline{z} - K^T y$ is the general solution of a nonhomogeneous system of linear $Az = b$, $\overline{z}$ is a particular solution of the system, and $K^T y$ is the general solution of a homogeneous system $Az = 0$.

Problem (7) can be written as a standard LP problem where the only restrictions are of the type:

$$\min_{y\in R^\upsilon}(c'y), \quad \overline{z} - K^T y \geqslant 0. \tag{8}$$

Recording (8) is sometimes called a problem with the same type of conditions or a conjugate canonical form [Моисеев, Иванилов, Столярова, 1978]. From a mathematical point of view, problem (8) is equivalent to the LP problem in canonical form:

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

Denote:

$\ker A$ is the null space (kernel) of the $(m \times n)$-matrix A of rank m:

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

$\operatorname{im} A$ is the space of rows of matrix A (image of matrix A^T):

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

The dimension of linear space $\ker A$ is equal to $\upsilon = n - m$, the defect of matrix A. The null space and the row space of a matrix A are orthogonal complements of each other. The space R^n is decomposed into the direct sum of these subspaces:

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

The rows of the matrix K are linearly independent by construction, belong to the null space of the matrix A, and therefore the space spanned by them $\operatorname{im} A^T$ coincides with the null space (kernel) of the matrix A. Thus, $\operatorname{im} K^T$ it is the orthogonal complement of the space $\operatorname{im} A^T$. Therefore,

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

Let us show that essentially (8) and (9) are the same problem. To do this, following Golikov and Evtushenko [Голиков, Евтушенко, 2000], for the $(n \times \upsilon)$-matrix K^T we define a pseudoinverse matrix $\left(K^T\right)^+$ and define the sets

$$Y = \left\{y \in R^\upsilon \mid \overline{z} - K^T y \geqslant 0\right\},$$
$$Z = \{z \in R^n \mid Az = b,\ z \geqslant 0\},$$
$$\overline{Z} = \{z \in R^n \mid Az = b\}.$$

The formula

$$z = \overline{z} - K^T y \tag{10}$$

can be considered as an affine mapping from R^υ to R^n. In this case, the image of a set Y is a set Z. There is a one-to-one correspondence between Z and Y. Indeed, for any $y \in Y$, formula (10) uniquely determines $z \in Z$. For an overdetermined system (10) of full rank, containing n linear equations and υ unknowns y, a pseudo-solution is always defined:

$$y(z) = \left(KK^T\right)^{-1} K(\overline{z} - z) = \left(K^T\right)^+ (\overline{z} - z), \tag{11}$$

which is the only solution to system (8) if and only if $\overline{z} - z \in \operatorname{im} K^T$. This inclusion takes place if and only if $z \in \overline{Z}$. So, for any, $z \in \overline{Z}$ formula (11) defines an affine transformation inverse to (10). Therefore we can write:

$$Y = \left(K^T\right)^+ (\overline{z} - Z). \tag{12}$$

Essentially, (8), (9) are the same problem. The external difference in the notation is associated with the change of variables (10), which made it possible, using formula (12), to transform the set Z into the set Y, which is the intersection n of half-spaces.

2.4. Geometric interpretation

First, following Moiseev, Ivanilov and Stolyarova [Моисеев, Иванилов, Столярова, 1978], we describe the geometric interpretation of problems of the form (9), then the geometric representation of the iterations of the simplex method and the direct algorithm.

Let us introduce an additional variable for (9):

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

and write it in the following form:

$$\min(z_0), \tag{13}$$

$$\begin{gathered} c''_1 z_1 + c''_2 z_2 + \dots + c''_n z_n = z_0, \\ a_{i1} z_1 + a_{i2} z_2 + \dots + a_{in} z_n = b_i, \quad i = 1, \dots, m, \end{gathered} \tag{14}$$

$$z_j \geqslant 0, \quad j = 1, \dots, n. \tag{15}$$

The right-hand side of equations (13) is a vector:

$$\widetilde{b}^j = \begin{pmatrix} z_0 & b_1 & \cdots & b_m \end{pmatrix}^T$$

with m fixed components and one variable component z_0. The left-hand side is a linear combination of the extended condition vectors:

$$\widetilde{a}^j = \begin{pmatrix} c''_j & ba_{1j} & \cdots & a_{mj} \end{pmatrix}^T, \quad j = 1, \dots, n,$$

with nonnegative coefficients z_j. Consider a set of $(m + 1)$ dimensional vectors:

$$u = \begin{pmatrix} u_0 & u_1 & \cdots & u_m \end{pmatrix}^T,$$

the components of which are determined by the relations:

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

Figure 1 shows this set for the case where $m = 2$, i. e., the problem has two equality type constraints.

Nonnegative linear combinations of extended condition vectors form a polyhedral cone $\widetilde{K}$, the edges of which will be the vectors $\widetilde{a}^1, \widetilde{a}^2, \dots, \widetilde{a}^5$. Their projections onto the plane $u_0 = 0$ are,

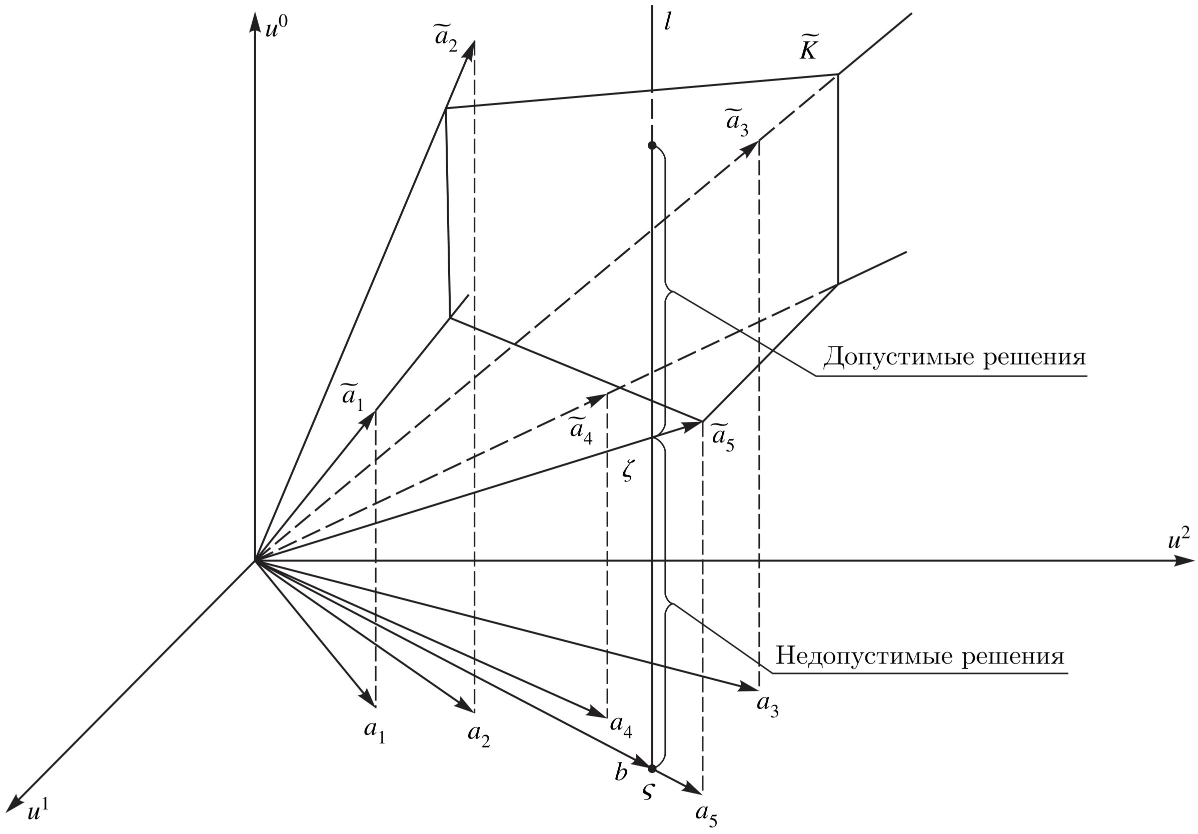

Рис. 1. Geometric interpretation

respectively, $a^1, a^2, \ldots, a^5$ (columns of the matrix of conditions A). The length of the perpendicular connecting the end of the vector $\widetilde{a}^j$ to the end of its projection a^j is equal to the linear form coefficient c''_j.

For an arbitrary value, z_0 the vector $\widetilde{b}$ points to a certain point lying on a vertical line l (see Fig. 1), passing through the end of the vector:

$$b = \begin{pmatrix} b_1 & b_2 & \cdots & b_m \end{pmatrix}^T$$

on the plane $u_0 = 0$ $\left(\text{point } \varsigma \text{ with coordinates } \begin{pmatrix} 0 & b_1 & \cdots & b_m \end{pmatrix}^T \text{ in Fig. 1}\right)$. If z_0 is a component of some admissible solution to problem (13)–(15), the vector $\widetilde{b}$ will also belong to the cone $\widetilde{K}$. It is also obvious that if a vector $\widetilde{b}$ points to the intersection point of a cone $\widetilde{K}$ and a line l, the corresponding value z_0 will be a component of some feasible solution. Thus, the admissible set of solutions to problem (13)–(15) is mapped onto a line segment l belonging to the cone $\widetilde{K}$. If the line l passes outside this cone, the problem has no feasible solutions. When the cone $\widetilde{K}$ contains the axis u_0, the value of the criterion on the admissible set is not limited from below, however, according to the construction of problem (7), this is impossible.

The values of the objective function of problem (13)–(15) on any of its admissible solutions corresponding to a certain vector $\widetilde{b}$ coincide and are equal to its first coordinate. Optimal solutions (there can be a whole set of them) are prototypes of the lower point of intersection of the straight line l with the cone $\widetilde{K}$ (point ζ in Fig. 1). The coefficients of an arbitrary decomposition of the vector $\widetilde{b}$ pointing to this point into a nonnegative linear combination of extended condition vectors $\widetilde{a}^j$ will be components of one of the optimal solutions. Obviously, among them, only components that correspond to vectors belonging to the same face of the cone $\widetilde{K}$ as the point ζ can be nonzero $\widetilde{a}^j$. If the number of such vectors does not exceed m and their corresponding vectors a^j are linearly independent, the problem has a unique solution. Otherwise there will be a continuum of solutions. Geometrically, the iterations of the simplex method represent a transition from one point to another in the direction of decreasing the objective function.

The forward algorithm generates a sequence of points outside the feasible region until a feasible solution is obtained. Thus, the points generated by the direct algorithm are prototypes of the points on a $\overline{\varsigma\zeta}$ line segment l (see Fig. 1), which do not belong to the cone $\widetilde{K}$, with the exception of the point ζ. Geometrically, the iterations of the direct algorithm represent a transition from one point to another in the direction of increasing the objective function.

3. The relation between the direct algorithm and the modified one using the simplex method and constructing an upper estimate for the number of iterations

To describe the relation, the computational scheme of the multiplicative algorithm of Malkov's simplex method was chosen [Малков, 1977], as the most suitable (describes the computational scheme of the second stage of the algorithm associated with the selection of a basic system of constraints, regardless of the method of its construction) in comparison with the known ones, for example, [Агафонова, Даугавет, 2017; Золотых, Кубарев, 2012; Малозёмов, 2017a].

3.1. The second stage of the simplex method algorithm

Let us relate the description of the computational scheme of the algorithm for solving problems of the form:

$$\max\left(\sum_{i=1}^{n} c_i x_i\right), \quad \sum_{i=1}^{n} a_{ji} x_i \leqslant b_j, \quad j = 1, \ldots, m, \quad x_i \geqslant 0, \quad i = 1, \ldots, n, \tag{16}$$

to the identification of the basic system of restrictions as follows. Let there be a basis: $\begin{pmatrix} x_{n_1} & \cdots & x_{n_m} \end{pmatrix}$ and a basis matrix $B_\nu = \begin{pmatrix} a_{\bullet n_1} & \cdots & a_{\bullet n_m} \end{pmatrix}$, as well as a multiplicative representation of the matrix B_ν^{-1}, where

$$B_\nu^{-1} = E_{r_\nu}^{\nu} E_{r_{\nu-1}}^{\nu-1} \cdots E_{r_1}^{1}, \quad E_{r_s}^{s} = \begin{pmatrix} 1 & & -\frac{x_{1k_s}}{x_{r_s k_s}} & & \\ & \ddots & \vdots & & \\ & & \frac{1}{x_{r_s k_s}} & & \\ & & \vdots & \ddots & \\ & & -\frac{x_{mk_s}}{x_{r_s k_s}} & & 1 \end{pmatrix}, \quad X_{k_s} = E_{r_{s-1}}^{s-1} \cdots E_{r_1}^{1} a_{k_s}.$$

Here r_s, k_s are the numbers of leading rows and columns at sth iteration of the algorithm. In order not to go into details of storing multipliers, we will recalculate the inverse basis matrix rather than present it as a product. Below is a detailed description of all operations performed in the course of solving problem (16) by the multiplicative algorithm of the simplex method.

Step 1. To calculate the current dual solution of the vector of row estimates using the formula:

$$u^\nu = c^\nu B_\nu^{-1},$$

where $c^\nu = \begin{pmatrix} c_{n_1} & \cdots & c_{n_m} \end{pmatrix}$ is the vector of prices of basic variables.

Step 2. Determine the leading column, the column to be entered into the basis:

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

$$d_k^1 = \min_i d_i^1 = \min_i \left(u^\nu a_{\bullet i} - c_i\right), \tag{17}$$

$$d_k^2 = \min_j d_j^2 = \min_j u_j^\nu. \tag{18}$$

If $d_k \geqslant \varepsilon_{\text{opt}}$, then the optimum has been achieved, where ε_{opt} is the minimum permissible value of the leading element, selected for reasons of stability of the computational process.

Step 3. Expand the leading column using the formula:

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

if the number k was determined from (17), and according to the formula:

$$X_k = B_\nu^{-1} E_k,$$

if the number k was determined from (18). Here E_k is the kth column of the identity matrix E.

Step 4. Calculate the leading line number from the relation:

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

where ε_{piv} is the minimum permissible value of the leading element, selected for reasons of stability of the computational process.

Step 5. Recalculate the basic solution:

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

recalculate the inverse basis matrix using the formula:

$$B_{\nu+1}^{-1} = E_{r_{\nu+1}}^{\nu+1} B_\nu^{-1},$$

where $E_{r_{\nu+1}}^{\nu+1}$ is the elementary matrix, multiplier:

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

Steps 1 to 5 are repeated until the optimum is reached, or until it turns out in step 4 that, $x_{jk} \leqslant \varepsilon_{\text{piv}}$, $j = 1, \ldots, m$, in other words, the solution to the problem is not limited.

The simplex method for solving LP problems, which uses the multiplicative representation of the inverse basis matrix as an algorithm for solving a system of equations, is mathematically attractive, but computationally vulnerable. As an example, consider a system consisting of one equation: $7x = 21$. The best way to solve this system is to divide: $x = \frac{21}{7} = 3$, but using the inverse matrix results in calculating: $x = \left(7^{-1}\right)21 = 2.99997$, which requires more arithmetic and produces a less accurate result. All of the above is also true for systems with many equations. Unnecessary actions are the main reason why this work focuses on direct multiplicative methods for solving systems, and not on methods of representing the inverse basis matrix.

When constructing a direct algorithm in terms of the simplex method, we will assume that at each iteration of the algorithm another one is added to the sequence of multipliers and that the size of this sequence is inconveniently large, but not precisely known.

3.2. The first stage of the direct algorithm

Let's consider the direct problem:

$$\min\left(\sum_{j=1}^{m} b_j u_j\right), \quad l_i(u) = \sum_{j=1}^{m} a_{ji} u_j - c_i \geqslant 0, \quad i = 1, \ldots, n, \quad u_i \geqslant 0, \quad j = 1, \ldots, m, \tag{19}$$

i. e., the dual problem (16). In the process of solving (19), an arbitrarily large scalar quantity is involved c_0; it is introduced to reduce compensation errors (see Example 1 in § 4).

Step 0. Put $\nu = 0$ and determine the number of the leading line using the formula:

$$\theta = x_{\bar{r}\nu} = \min_j x_{j\nu} = \min_j b_j.$$

If $\theta < 0$, then calculate the initial dual solution of the vector of row estimates using the formula:

$$u_j^\nu = 0, \quad j \neq \bar{r}, \quad u_{\bar{r}}^\nu = -\frac{c_0}{x_{\bar{r}\nu}},$$

calculate the inverse basis matrix using the formula:

$$B_\nu^{-1} = \begin{pmatrix} 1 & & -\frac{x_{1\nu}}{x_{\bar{r}\nu}} & & \\ & \ddots & \vdots & & \\ & & \frac{1}{x_{\bar{r}\nu}} & & \\ & & \vdots & \ddots & \\ & & -\frac{x_{m\nu}}{x_{\bar{r}\nu}} & & 1 \end{pmatrix},$$

set $\nu = 1$ and go to step 1 of the first stage. If $\theta \geqslant 0$, then calculate the dual solution of the vector of row estimates using the formula: $u_j^0 = 0$, calculate the inverse basis matrix using the formula: $B_0^{-1} = E$, where E is the identity matrix, calculate the basic solution using the formula: $\bar{x}_j = x_{j0}$, put $\nu = 1$ and go to step 1 of the second stage.

Step 1. Determine the leading column, the column to be entered into the basis:

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

$$d_k^1 = \min_i d_i^1 = \min_i \left(u^\nu a_{\bullet i} - c_i\right), \tag{20}$$

$$d_k^2 = \min_j d_j^2 = \min_j u_j^\nu. \tag{21}$$

If $d_k \geqslant \varepsilon_{\text{opt}}$, then the solution to the problem is unlimited.

Step 2. Expand the leading column using the formula:

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

if the number k is determined from (20), and by the formula:

$$X_k = B_\nu^{-1} E_k,$$

if the number k is determined from (21).

Step 3. Calculate the leading line number from the relation:

$$\theta = x_{rk} = \min_{j,\, j \neq \bar{r}} x_{jk}.$$

If $\theta \leqslant \varepsilon_{\text{piv}}$, $x_{\bar{r}k} \leqslant \varepsilon_{\text{piv}}$, then the conditions of the problem are inconsistent; if $\theta \leqslant \varepsilon_{\text{piv}}$, $x_{\bar{r}k} > \varepsilon_{\text{piv}}$, then set $r = \bar{r}$ and go to step 4; if $\theta > \varepsilon_{\text{piv}}$, then go to step 4.

Step 4. Recalculate the current dual solution of the vector of row estimates:

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

where E_r is the rth column of the identity matrix.

Step 5. Recalculate the inverse basis matrix:

$$B_{\nu+1}^{-1} = E_{r_{\nu+1}}^{\nu+1} B_\nu^{-1},$$

where $E_{r_{\nu+1}}^{\nu+1}$ is the elementary matrix, multiplier:

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

Step 6. If $r \neq \bar{r}$, then set $\nu = \nu + 1$ and go to step 1 of the first stage; otherwise calculate the basic solution:

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

set $\nu = \nu + 1$ and go to step 1 of the second stage.

Steps 1 to 6 are repeated until it becomes clear that:

- the solution to the problem is unlimited (step 1);
- the conditions of the problem are inconsistent (step 3);
- the condition for transition to step 1 of the second stage (step 6) has been fulfilled.

3.3. The second stage of the direct algorithm

Step 1. Determine the leading column, the column to be entered into the basis:

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

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

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

If $d_k \geqslant \varepsilon_{\text{opt}}$, then the optimum has been achieved.

Step 2. Expand the leading column using the formula:

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

if the number k was determined from (22), and according to the formula:

$$X_k = B_\nu^{-1} E_k,$$

if the number k was determined from (23).

Step 3. Calculate the leading line number from the relation:

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

If $x_{jk} \leqslant \varepsilon_{\text{piv}}$, $j = 1, \ldots, m$, then the conditions of the problem are inconsistent.

Step 4. Recalculate the current dual solution of the vector of row estimates:

$$u^{\nu+1} = u^{\nu} - \frac{d_k}{x_{rk}} (E_r)^T B_\nu^{-1},$$

where E_r is the rth column of the identity matrix.

Step 5. Recalculate the basic solution:

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

recalculate the inverse basis matrix:

$$B_{\nu+1}^{-1} = E_{r_{\nu+1}}^{\nu+1} B_\nu^{-1},$$

where $E_{r_{\nu+1}}^{\nu+1}$ is the elementary matrix, multiplier:

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

Steps 1 to 5 are repeated until the optimum is reached, or until it turns out in step 3 that the conditions of the problem are inconsistent.

3.4. Comparison of algorithms

Let us prove that for the same initial data (inputs), the algorithms generate the same sequence of points: the basic solution and the current dual solution of the vector of row estimates.

Statement 1. *Second stage algorithms for the same initial data generate the same sequence of points.*

Proof. The statement is true if and only if:

$$c^{\nu+1} E_{r_{\nu+1}}^{\nu+1} B_\nu^{-1} = u^\nu - \frac{d_k}{x_{rk}} E_r^T B_\nu^{-1}.$$

Assuming $u^\nu = c^\nu B_\nu^{-1}$, we get:

$$c^{\nu+1} E_{r_{\nu+1}}^{\nu+1} = c^\nu - \frac{d_k}{x_{rk}} E_r^T.$$

Since $c_j^{\nu+1} = c_j^\nu$, $j \neq r$, it remains to prove that:

$$-\sum_{j=1,\ j\neq r}^{m} \frac{c_j^{\nu+1} x_{jk}}{x_{rk}} + \frac{c_r^{\nu+1}}{x_{rk}} = c_r^\nu - \frac{d_k}{x_{rk}}.$$

There are two possible cases. In the first, the number k was determined from (20), and in the second from (21).

In the first case:

$$c_r^\nu - \frac{d_k}{x_{rk}} = c_r^\nu - \frac{u^\nu a_{\bullet k} - c_k^\nu}{x_{rk}} = c_r^\nu - \frac{c^\nu B_\nu^{-1} a_{\bullet k} - c_k^\nu}{x_{rk}} =$$

$$= c_r^\nu - \frac{c^\nu \begin{pmatrix} x_{1k} & \cdots & x_{mk} \end{pmatrix}^T - c_k^\nu}{x_{rk}} = - \sum_{j=1,\ j\neq r}^{m} \frac{c_j^{\nu+1} x_{jk}}{x_{rk}} + \frac{c_r^{\nu+1}}{x_{rk}}.$$

In the second case:

$$c_r^\nu - \frac{d_k}{x_{rk}} = c_r^\nu - \frac{u_k^\nu}{x_{rk}} = c_r^\nu - \frac{u^\nu E_k}{x_{rk}} = c_r^\nu - \frac{c^\nu B_\nu^{-1} E_k}{x_{rk}} =$$

$$= c_r^\nu - \frac{c^\nu \begin{pmatrix} x_{1k} & \cdots & x_{mk} \end{pmatrix}^T}{x_{rk}} = - \sum_{j=1,\ j\neq r}^{m} \frac{c_j^\nu x_{jk}}{x_{rk}} = - \sum_{j=1,\ j\neq r}^{m} \frac{c_j^{\nu+1} x_{jk}}{x_{rk}} + \frac{c_r^{\nu+1}}{x_{rk}}.$$

The statement has been proven. □

From this, in particular, it follows that the direct algorithm is an algorithm like the simplex method.

When comparing algorithms, we assume that the inverse matrix B_ν^{-1} is completely filled. Obviously, the difference in the number of arithmetic operations occurs at step 1 of the simplex method algorithm and step 4 of the direct algorithm. It follows that the gain in the number of arithmetic operations of multiplication and addition in one iteration is equal to $m^2 - m - 1$.

3.5. The number of iterations

Let us consider problem (6) in the form of an LP problem with restrictions of only the type of inequalities:

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

By virtue of construction (6), problems of the form (24) satisfy the conditions:

- $c' \geqslant 0$, $\mp b \ngeq 0$;
- the rows in the matrices N are linearly independent by constructing equations for connecting $z_i = b_i + e^\nu_{\nu_i \bullet} y$ of the variables $z_i \geqslant 0$, and the rows in the matrix in $N' = \begin{pmatrix} c' \\ N \end{pmatrix}$, by the assumption of the uniqueness of the solution to problem (1), which is equivalent to (6);
- there are either no υ or fewer linearly dependent rows in the matrix $\begin{pmatrix} N' \\ I_\upsilon \end{pmatrix}$ or $\upsilon + 1$ or fewer linearly dependent rows in $\begin{pmatrix} b & N \\ 0_\upsilon & I_\upsilon \end{pmatrix}$.

In particular, the LP problem in this model will be nondegenerate. Let's denote:

$p = -b^T$ is the m-dimensional row vector,

$G = \begin{pmatrix} N & -I_m \end{pmatrix}^T$ is the $(n \times m)$ matrix of rank m in block form,

$g = \begin{pmatrix} c' & 0_m^T \end{pmatrix}^T$ is the n-dimensional vector in block form.

With the introduced notation, the problem dual to (24) takes the form:

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

We limit ourselves to deriving an upper bound $\frac{m}{2}$ for problems of the form (25), which is consistent with the almost linear increase in the number of iterations of the simplex method in practice. The model is natural. Along with a linear program of the form (25), an initial vertex u^ν (the initial value of the iteration counter $\nu = 0$) is given. Then choose a row vector $\overline{p}$ that satisfies the condition:

$$\overline{p}u^\nu = \max_{u \in R^m}(\overline{p}u), \quad Gu \leqslant g. \tag{26}$$

REMARK. For an arbitrary vertex u^ν it is easy to find some row vector $\overline{p}$ satisfying condition (26), for example, you can add rows G in which equality is achieved in the system $Gu^\nu \leqslant g$ [Схрейвер, 1991]. The function $u \to \overline{p}u$ is called a co-objective function.

Then one of the rules for calculating the leading line is selected: Danzig or according to the largest contribution to the change in the value of the objective function. The first rule is simpler, however, the choice of the second when solving ill-conditioned problems leads to an increase in the accuracy of the solution, and when sequences of iterations arise in which the objective function practically does not change, to a significant reduction in the number of iterations. Thus, the mechanism for selecting the leading line can be described as follows. When solving ill-conditioned problems, the second rule is chosen, otherwise the first one is chosen. If the second rule is selected, the selection mechanism is described. Otherwise, the choice of the leading line is ensured by the internal structure of the algorithm as follows. At each iteration of the algorithm, the increment of the objective function is estimated. If it is less than a given threshold value, then the second rule is selected, otherwise the first one.

The direct algorithm for solving the nondegenerate problem (24) and the simplex method algorithm for solving (25) generate (from Statement 1) the same sequence of points (the basic solution and the current dual solution of the vector of row estimates). Moreover, geometrically, the iterations of the direct algorithm represent a transition from one point to another in the direction of increasing the objective function. Thus, the mechanism for selecting the leading line satisfies the conditions:

(i) the problem (25) is degenerated;
(ii) geometrically the iterations of the simplex method represent a transition from one point u^ν to another $u^{\nu+1}$ in the direction of increasing objective function $u \to pu$, therefore in the direction of decreasing the co-objective function $u \to \overline{p}u$;
(iii) for any fixed $\overline{p}$ reversing the signs of some inequalities, in which equality is achieved in the system $Gu^\nu \leqslant g$ (so that the maximum $\overline{p}u$ on the modified polyhedron is finite) or replacing p to $-p$ does not depend on G, g, p, $\overline{p}$. (27)

The practical meaning of conditions (27) can be questioned if one does not take into account ill-conditioned problems that arise in practice with strong instability of numerical solutions regarding errors in source data or machine rounding. However, this is, first of all, the subjective point of view of the author.

REMARK (RATIONALE). Without loss of generality, we will assume that at the next iteration of the algorithm a system with a Hilbert matrix is solved. These matrices are interesting because they are poorly conditioned even for relatively small values of their order. This feature allows them to be used for testing and debugging various computational algorithms. Systems with ill-conditioned matrices are usually called unstable or ill-conditioned. In general, they are characterized by the fact that a slight change in the conditions of the account can lead to unacceptably large errors in the solution. Moreover, it is known that the use of “long number” mathematics makes it possible to obtain the main characteristics of Hilbert matrices (for example, the values of norms, condition numbers, etc.) up to the hundredth order and higher [Майстренко, Светлаков, Черепанов, 2011]. It is impossible in principle to obtain similar results using single, double or maximum machine precision. Thus, constructing an algorithm for solving such large-scale problems using single, double, maximum machine precision or "long number" mathematics is also impossible.

Now one can carry out the leading operation and find $u^{\nu+1}$.

Since (from (27), (i)) $pu^{\nu+1} \neq pu^{\nu}$, $\overline{p}u^{\nu+1} \neq \overline{p}u^{\nu}$ and (from (27), (ii)) $pu^{\nu+1} > pu^{\nu}$, $\overline{p}u^{\nu+1} < \overline{p}u^{\nu}$ and the programm (25) is unlimited, we have:

$$pu^{\nu+1} > pu^{\nu}, \quad \overline{p}u^{\nu+1} < \overline{p}u^{\nu}, \quad p\varphi = \left(u^{\nu+1} - u^{\nu}\right) > 0, \quad \overline{p}\varphi < 0. \tag{28}$$

Statement 2. *If the leading row selection mechanism satisfies* (27)*, then the simplex method solves problems of the form* (25) *in no more than* $\frac{m}{2}$ *iterations.*

Proof. Suppose we are given problem (25), a vertex u^{ν} of a polyhedron P and a row vector $\overline{p}$, for which u^{ν} maximizes $\overline{p}u$ on the set $Gu \leqslant g$. Let n and m be the number of rows and columns of the matrix G, respectively.

Let $\omega^1, \ldots, \omega^t$ be vectors from R^n, which are solutions of subsystems of n linearly independent equations of the system $Gu = g$. Thus, $t = C_n^m$.

Let us consider the class L of LP problems of the form:

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

formed from program (25) by inverting some (possibly empty) set of inequalities and (or) replacing the vector p by $-p$, for which the maximum is finite:

$$\max_{u\in R^m}(\overline{p}u), \quad \widetilde{G}u \leqslant \widetilde{g}. \tag{29}$$

Let's show that $|L| = 2C_n^m$. Indeed, from the condition of Statement 1 it follows that each maximum (29) is achieved at a single vertex $\widetilde{G}u \leqslant \widetilde{g}$ and, therefore, at one of the vectors ω^s. Moreover, for each ω^s there is exactly one choice of system $\widetilde{G}u \leqslant \widetilde{g}$, for which the maximum (29) is achieved at ω^s since exactly n inequalities from the system $\widetilde{G}u \leqslant \widetilde{g}$, say $\widetilde{G}'u \leqslant \widetilde{g}'$, are satisfied ω^s as equalities. Thus, in the remaining $n - m$ inequalities, the inequality sign during substitution ω^s will be strict and therefore cannot be reversed. Moreover, the row vector $\overline{p}$ must belong to the cone generated by the rows $\widetilde{G}$ and, therefore, the signs are uniquely defined here. Thus, to ω^s there correspond the programs:

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

Hence, $|L| = 2C_n^m$.

Now consider the set of all finite and infinite edges of all polyhedra defined by systems $\widetilde{G}u \leqslant \widetilde{g}$. Each set of $m - 1$ equations of the system $Gu = g$ generates a line (one-dimensional affine subspace) in R^m containing $n - m + 1$ points from ω^s, divide it into $n - m$ edge-segments and two infinite edges-rays. Let l be such a straight, and let ω^s and ω^μ be two neighboring points on l, i. e., suppose that the segment $\overline{\omega^s\omega^\mu}$ contains no other points from $\omega^1, \ldots, \omega^t$. Then the edge $\overline{\omega^s\omega^\mu}$ is traversed in one of the class programs exactly once. Let's assume the opposite, namely, that $\overline{\omega^s\omega^\mu}$ is traversed in more than one program:

$$\max_{u\in R^m}(\widetilde{p}u), \quad \widetilde{G}u \leqslant \widetilde{g}.$$

For a point ω^s we find a row vector $\overline{p}$ satisfying the condition:

$$\overline{p}\omega^s = \max_{u\in R^m}(\overline{p}u), \quad u \in \widetilde{P} = \{u \mid \widetilde{G}u \leqslant \widetilde{g}\},$$

and for the polyhedron $\widetilde{P}$ the following inequality holds:

$$\overline{p}u \leqslant \overline{p}\omega^s. \tag{30}$$

From (27) it follows that exactly $m-1$ from the inequalities of the system $\widetilde{G}u \leqslant \widetilde{g}$ turn into equalities on $\overline{\omega^s\omega^\mu}$. Let's denote the matrix of coefficients of the subsystem of such inequalities of rank $m-1$ by $\widetilde{G}_{eq(\overline{\omega^s\omega^\mu})}$. Thus, the inequality signs in other inequalities of the system cannot be reversed. Moreover, there is exactly one $(m-1)$-dimensional row vector w satisfying $w\widetilde{G}_{eq(\overline{\omega^s\omega^\mu})} = \overline{p}$. Since inequality (30) specifies the support subspace, it follows that $w^T > 0$. The uniqueness of w implies the uniqueness of inequality signs in $\widetilde{G}u \leqslant \widetilde{g}$. Finally, if $p\omega^s > p\omega^\mu$, then it follows from (28) that $\widetilde{p} = p$. If $p\omega^s < p\omega^\mu$, then it follows from (28) that $\widetilde{p} = -p$. Thus, for $\widetilde{p}$ there is also a unique choice, so that the edge $\overline{\omega^s\omega^\mu}$ is traversed in the only program from L.

Similarly, it follows from (28) that each ray $\omega^s + \varphi R_+$ is traversed only if $\overline{p}\varphi > 0$, and therefore only in one program from L.

On each straight line defined by $m-1$ equations from the system $Gu = g$ there are $n-m$ edges and one ray $\omega^s + \varphi R_+$ with $\overline{p}\varphi > 0$. Thus, the total number of such rays and edges is equal to $(n-m+1)C_n^{m-1} = mC_n^m$, so the number of iterations is no more than $\frac{mC_n^m}{|L|} = \frac{1}{2}m$.

This proves the statement. □

In particular, it follows from this and Statement 1 that the direct algorithm solves problems of the form (24) in no more than $\frac{m}{2}$ iterations.

4. Examples of problem solutions with guaranteed behavior of the simplex method

Let's consider a direct algorithm using examples of solutions to problems with guaranteed behavior of the simplex method in a phenomenological sense (the sense of understanding the phenomenon) and in a constructive sense (how the thing to be defined is structured and how to work with it) [Лачинов, Поляков, 1999].

Example 1 (Shevchenko – Zolotykh [Шевченко, Золотых, 2002]). Find a solution to problem (1) for looping the simplex method with Dantzig's rule:

$$c = \begin{pmatrix} 0 & 0 & 0 & 0 & -200 & -175 & 1100 & 2 \end{pmatrix},$$

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

Let us show that the solution to the Shevchenko – Zolotykh problem is achieved at the first stage of the algorithm. The following auxiliary quantities are involved in the solution process:

c_0 $\left(-c_0 \leqslant cx^*\right)$ is an arbitrarily large scalar quantity which is introduced to reduce compensation errors;

x_0 $(x_0 \geqslant 0)$ is an additional variable;

$c_0 - x_0 + cx = 0$ is the constraint equation of an additional variable;

$\min\limits_{x\in R^n}(x_0)$, $c_0 - x_0 + cx = 0$, $h + Hx = 0$, $x_0 \geqslant 0$, $x \geqslant 0$, is a problem equivalent to the Shevchenko – Zolotykh problem.

REMARK. Among arithmetic operations there are those that can lead to relative errors that exceed the value of machine precision many times, for example, calculating the differences of almost identical rounded numbers. The errors associated with them are usually called compensation errors [Гилл, Мюррей, Райт, 1985].

Step 0 (initialization). Calculate the number r of the leading column (the number of the leading row $q = 0$) using the formula:

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

Rewrite the constraint equation of an additional variable $x_0 \geqslant 0$:

$$x_5 = x_5^0 + e^0_{5_0\bullet}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_0 & x_6 & x_7 & x_8 \end{pmatrix}^T,$$

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

Set:

$$x^0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{200}c_0 & 0 & 0 & 0 \end{pmatrix}^T.$$

Step 1 (calculation x^1). Calculate the elements of the constraint equation of the fourth row (select the number of the leading row $q = 4$):

$$h_4^1 + \begin{pmatrix} h_{41}^1 & \cdots & h_{48}^1 \end{pmatrix}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_0 & x_6 & x_7 & x_8 \end{pmatrix}^T = 0,$$
$$h_4^1 = h_4 + h_{45}x_5^0 = 1, \quad \begin{pmatrix} h_{41}^1 & \cdots & h_{48}^1 \end{pmatrix} = h_{4\bullet}E^0_{5_0} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Rewrite the constraint equation of the fourth row (to simplify the calculations, we choose the number of the leading column to be $r = 8$):

$$x_8 = x_8^1 + e^1_{8_1\bullet}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_0 & x_6 & x_7 \end{pmatrix}^T,$$
$$x_8^1 = 1, \quad e^1_{8_1\bullet} = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix},$$
$$E^1_{8_1} = \begin{pmatrix} 1 &&&&&& \\ & 1 &&&&& \\ && 1 &&&& \\ &&& 1 &&& \\ &&&& 1 && \\ &&&&& 1 & \\ &&&&&& 1 \\ &&& -1 &&& \end{pmatrix}.$$

Recalculate the elements of the constraint equation:

$$x_5 = x_5^1 + e^0_{5_1\bullet}\begin{pmatrix}x_1 & x_2 & x_3 & x_4 & x_0 & x_6 & x_7\end{pmatrix}^T,$$

$$x_5^1 = \frac{1}{100} + \frac{1}{200}c_0, \quad e^0_{5_1\bullet} = e^0_{5_0\bullet}E^1_{8_1} = \begin{pmatrix}0 & 0 & 0 & -\tfrac{1}{100} & -\tfrac{1}{200} & -\tfrac{7}{8} & \tfrac{11}{2}\end{pmatrix},$$

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

Set:

$$x^1 = \begin{pmatrix}0 & 0 & 0 & 0 & \tfrac{1}{100} + \tfrac{1}{200}c_0 & 0 & 0 & 1\end{pmatrix}^T.$$

Step 2 (calculation x^2). Calculate the elements of the constraint equation of the first line (select $q = 1$):

$$h_1^2 + \begin{pmatrix}h^2_{11} & h^2_{12} & \cdots & h^2_{17}\end{pmatrix}\begin{pmatrix}x_1 & x_2 & x_3 & x_4 & x_0 & x_6 & x_7\end{pmatrix}^T = 0,$$

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

$$\begin{pmatrix}h^2_{11} & h^2_{12} & \cdots & h^2_{17}\end{pmatrix} = h_{1\bullet}E^0_{5_0}E^1_{8_1} = \begin{pmatrix}0 & 1 & 0 & \tfrac{1}{100} & \tfrac{3}{200} & \tfrac{11}{8} & -\tfrac{19}{2}\end{pmatrix}.$$

Rewrite the constraint equation of the first line (select $r = 2$):

$$x_2 = x_2^2 + e^2_{2_2\bullet}\begin{pmatrix}x_1 & x_3 & x_4 & x_0 & x_6 & x_7\end{pmatrix}^T,$$

$$x_2^2 = \frac{1}{100} + \frac{3}{200}c_0, \quad e^2_{2_2\bullet} = \begin{pmatrix}0 & 0 & -\tfrac{1}{100} & -\tfrac{3}{200} & -\tfrac{11}{8} & \tfrac{19}{2}\end{pmatrix},$$

$$E^2_{2_2} = \begin{pmatrix}1 & & & & & \\ & & -\tfrac{1}{100} & -\tfrac{3}{200} & -\tfrac{11}{8} & \tfrac{19}{2} \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1\end{pmatrix}.$$

Recalculate the elements of the constraint equations:

$$x_8 = x_8^2 + e^1_{8_2\bullet}\begin{pmatrix}x_1 & x_3 & x_4 & x_0 & x_6 & x_7\end{pmatrix}^T,$$

$$x_8^2 = 1, \quad e^1_{8_2\bullet} = e^1_{8_1\bullet}E^2_{2_2} = \begin{pmatrix}0 & 0 & -1 & 0 & 0 & 0\end{pmatrix},$$

$$x_5 = x_5^2 + e^0_{5_2\bullet}\begin{pmatrix}x_1 & x_3 & x_4 & x_0 & x_6 & x_7\end{pmatrix}^T,$$

$$x_5^2 = \frac{1}{100} + \frac{1}{200}c_0, \quad e^0_{5_2\bullet} = e^0_{5_1\bullet}E^2_{2_2} = \begin{pmatrix}0 & 0 & -\tfrac{1}{100} & -\tfrac{1}{200} & -\tfrac{7}{8} & \tfrac{11}{2}\end{pmatrix},$$

$$E^0_{5_0}E^1_{8_1}E^2_{2_2}=\begin{pmatrix}1&&&&\\&-\frac{1}{100}&-\frac{3}{200}&-\frac{11}{8}&\frac{19}{2}\\&1&&&\\&&1&&\\&-\frac{1}{100}&-\frac{1}{200}&-\frac{7}{8}&\frac{11}{2}\\&&&1&\\&&&&1\\&-1&&&\end{pmatrix}.$$

Set:

$$x^2=\begin{pmatrix}0&\frac{1}{100}+\frac{3}{200}c_0&0&0&\frac{1}{100}+\frac{1}{200}c_0&0&0&1\end{pmatrix}^T.$$

Step 3 (calculation x^3). Calculate the elements of the constraint equation of the second line (select $q=2$):

$$h^3_2+\begin{pmatrix}h^3_{21}&h^3_{22}&\cdots&h^3_{26}\end{pmatrix}\begin{pmatrix}x_1&x_3&x_4&x_0&x_6&x_7\end{pmatrix}^T=0,$$
$$h^3_2=h_2+h_{22}x^2_2+h_{25}x^2_5=\frac{1}{60}-\frac{1}{600}c_0,$$
$$\begin{pmatrix}h^3_{21}&h^3_{22}&\cdots&h^3_{26}\end{pmatrix}=h_{2\bullet}E^0_{5_0}E^1_{8_1}E^2_{2_2}=\begin{pmatrix}-1&0&-\frac{1}{60}&\frac{1}{600}&\frac{1}{8}&-\frac{5}{6}\end{pmatrix}.$$

Rewrite the constraint equation of the second line (select $r=5$):

$$x_6=x^3_6+e^3_{6_3\bullet}\begin{pmatrix}x_1&x_3&x_4&x_0&x_7\end{pmatrix}^T,$$
$$x^3_6=-\frac{2}{15}+\frac{1}{75}c_0,\quad e^3_{6_3\bullet}=\begin{pmatrix}8&0&\frac{2}{15}&-\frac{1}{75}&\frac{20}{3}\end{pmatrix},$$
$$E^3_{6_3}=\begin{pmatrix}1&&&&\\&1&&&\\&&1&&\\&&&&\\&&&1&\\8&0&\frac{2}{15}&-\frac{1}{75}&\frac{20}{3}\\&&&&1\end{pmatrix}.$$

Recalculate the elements of the constraint equations:

$$x_8=x^3_8+e^1_{8_3\bullet}\begin{pmatrix}x_1&x_3&x_4&x_0&x_7\end{pmatrix}^T,$$
$$x^3_8=1,\quad e^1_{8_3\bullet}=e^1_{8_2\bullet}E^3_{6_3}=\begin{pmatrix}0&0&-1&0&0\end{pmatrix},$$
$$x_5=x^3_5+e^0_{5_3\bullet}\begin{pmatrix}x_1&x_3&x_4&x_0&x_7\end{pmatrix}^T,$$
$$x^3_5=\frac{19}{150}-\frac{1}{150}c_0,\quad e^3_{5_3\bullet}=e^3_{5_2\bullet}E^3_{6_3}=\begin{pmatrix}-7&0&\frac{19}{150}&\frac{1}{150}&-\frac{1}{3}\end{pmatrix},$$
$$x_2=x^3_2+e^2_{2_3\bullet}\begin{pmatrix}x_1&x_3&x_4&x_0&x_7\end{pmatrix}^T,$$
$$x^3_2=\frac{29}{150}-\frac{1}{300}c_0,\quad e^2_{2_3\bullet}=e^2_{2_2\bullet}E^3_{6_3}=\begin{pmatrix}-11&0&-\frac{29}{150}&\frac{1}{300}&\frac{1}{3}\end{pmatrix},$$

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

Set:

$$x^3=\begin{pmatrix}0&\frac{29}{150}-\frac{1}{300}c_0&0&0&\frac{19}{150}-\frac{1}{150}c_0&-\frac{2}{15}+\frac{2}{150}c_0&0&1\end{pmatrix}^T.$$

Step 4 (calculation x^4). Calculate the elements of the constraint equation of the third row:

$$h_3^4+\begin{pmatrix}h_{31}^4&h_{32}^4&\cdots&h_{35}^4\end{pmatrix}\begin{pmatrix}x_1&x_3&x_4&x_0&x_7\end{pmatrix}^T=0,$$

$$h_3^4=h_3+h_{32}x_2^3+h_{35}x_5^3+h_{36}x_6^3+h_{38}x_8^3=\frac{35}{6}-\frac{1}{3}c_0,$$

$$\begin{pmatrix}h_{31}^4&h_{32}^4&\cdots&h_{35}^4\end{pmatrix}=h_{3\bullet}E_{5_0}^{0}E_{8_1}^{1}E_{2_2}^{2}E_{6_3}^{3}=\begin{pmatrix}-\frac{625}{2}&1&-\frac{35}{6}&\frac{1}{3}&-\frac{325}{6}\end{pmatrix}.$$

Rewrite the constraint equation of the third line ($r=2$):

$$x_3=x_3^4+e_{3_4\bullet}^4\begin{pmatrix}x_1&x_4&x_0&x_7\end{pmatrix}^T,$$

$$x_3^4=-\frac{35}{6}+\frac{1}{3}c_0,\quad e_{3_4\bullet}^4=\begin{pmatrix}\frac{625}{2}&\frac{35}{6}&-\frac{1}{3}&\frac{325}{6}\end{pmatrix},$$

$$E_{3_4}^4=\begin{pmatrix}1&&&\\\frac{625}{2}&\frac{35}{6}&-\frac{1}{3}&\frac{325}{6}\\&1&&\\&&1&\\&&&1\end{pmatrix}.$$

Recalculate the elements of the constraint equations:

$$x_5=x_5^4+e_{5_4\bullet}^0\begin{pmatrix}x_1&x_4&x_0&x_7\end{pmatrix}^T,$$

$$x_5^4=\frac{19}{150}-\frac{1}{150}c_0,\quad e_{5_4\bullet}^0=e_{5_3\bullet}^0E_{3_4}^4=\begin{pmatrix}-7&-\frac{19}{150}&\frac{1}{150}&-\frac{1}{3}\end{pmatrix},$$

$$x_6=x_6^4+e_{6_4\bullet}^3\begin{pmatrix}x_1&x_4&x_0&x_7\end{pmatrix}^T,$$

$$x_6^4=-\frac{2}{15}+\frac{1}{75}c_0,\quad e_{6_4\bullet}^3=e_{6_3\bullet}^3E_{3_4}^4=\begin{pmatrix}8&\frac{2}{15}&-\frac{2}{150}&\frac{20}{3}\end{pmatrix},$$

$$x_2=x_2^4+e_{2_4\bullet}^2\begin{pmatrix}x_1&x_4&x_0&x_7\end{pmatrix}^T,$$

$$x_2^4=\frac{29}{150}-\frac{1}{300}c_0,\quad e_{2_4\bullet}^2=e_{2_3\bullet}^2E_{3_4}^4=\begin{pmatrix}-11&-\frac{29}{150}&\frac{1}{300}&\frac{1}{3}\end{pmatrix},$$

$$x_8=x_8^4+e_{8_4\bullet}^1\begin{pmatrix}x_1&x_4&x_0&x_7\end{pmatrix}^T,$$

$$x_8^4=1,\quad e_{8_4\bullet}^1=e_{8_3\bullet}^1E_{3_4}^4=\begin{pmatrix}0&-1&0&0\end{pmatrix}.$$

Rewrite the constraint equation of the variable $x_5 \geqslant 0$ $\left(x_5^4 < 0\right)$:

$$x_0 = x_0^4 + e_{0_4\bullet}^4 \begin{pmatrix} x_1 & x_4 & x_5 & x_7 \end{pmatrix}^T,$$

$$x_0^4 = -19 + c_0, \quad e_{0_4\bullet}^4 = \begin{pmatrix} 1050 & 19 & 150 & 50 \end{pmatrix},$$

$$E_{0_4}^4 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1050 & 19 & 150 & 50 \\ & & & 1 \end{pmatrix}.$$

Recalculate the elements of the constraint equations:

$$x_2 = x_2^5 + e_{2_5\bullet}^2 \begin{pmatrix} x_1 & x_4 & x_5 & x_7 \end{pmatrix}^T,$$

$$x_2^5 = \frac{13}{100}, \quad e_{2_5\bullet}^2 = e_{2_4\bullet}^2 E_{0_4}^4 = \begin{pmatrix} -\frac{15}{2} & -\frac{13}{100} & \frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

$$x_3 = x_3^5 + e_{3_5\bullet}^4 \begin{pmatrix} x_1 & x_4 & x_5 & x_7 \end{pmatrix}^T,$$

$$x_3^5 = \frac{1}{2}, \quad e_{3_5\bullet}^4 = e_{3_4\bullet}^4 E_{0_4}^4 = \begin{pmatrix} -\frac{75}{2} & -\frac{1}{2} & -50 & \frac{75}{2} \end{pmatrix},$$

$$x_6 = x_6^5 + e_{6_5\bullet}^3 \begin{pmatrix} x_1 & x_4 & x_5 & x_7 \end{pmatrix}^T,$$

$$x_6^5 = \frac{3}{25}, \quad e_{6_5\bullet}^3 = e_{6_4\bullet}^3 E_{0_4}^4 = \begin{pmatrix} -6 & -\frac{3}{25} & -2 & 6 \end{pmatrix},$$

$$x_8 = x_8^5 + e_{8_5\bullet}^1 \begin{pmatrix} x_1 & x_4 & x_5 & x_7 \end{pmatrix}^T,$$

$$x_8^5 = 1, \quad e_{8_5\bullet}^1 = e_{8_4\bullet}^1 E_{0_4}^4 = \begin{pmatrix} 0 & -1 & 0 & 0 \end{pmatrix}.$$

Set:

$$x^* = x^4 = \begin{pmatrix} 0 & \frac{13}{100} & \frac{1}{2} & 0 & 0 & \frac{3}{25} & 0 & 1 \end{pmatrix}^T \geqslant 0, \quad cx^* = -19.$$

REMARK. Compensation errors are minimized by eliminating an additional variable x_0 in the process of calculating x^4.

EXAMPLE 2 (SHEVCHENKO – ZOLOTYKH [ШЕВЧЕНКО, ЗОЛОТЫХ, 2002]). Find a solution to the simplex method looping problem with Dantzig's rule:

$$c = \begin{pmatrix} 600 & 0 & 0 & 0 & 0 & -75 & 500 & -2 \end{pmatrix},$$

$$H = \begin{pmatrix} 9 & 1 & 0 & 0 & 0 & \frac{1}{4} & -2 & -\frac{1}{25} \\ 3 & 0 & 1 & 0 & 0 & \frac{1}{2} & -3 & -\frac{1}{50} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -\frac{225}{2} & 0 & 0 & 0 & 1 & -25 & 200 & 1 \end{pmatrix}, \quad h = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Let us show that the solution to the Shevchenko – Zolotykh problem is achieved at the second stage of the algorithm. Below we will use the superscript ν as an iteration counter. The initial value of this counter is $\nu = 0$, and the problem constructed at the fourth step of the first stage of solving the Shevchenko – Zolotykh problem [Свириденко, 2017] can be rewritten as:

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

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

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

$$e_{0\bullet}^\nu = \begin{pmatrix} 1925 & 100 & 100 & 8 \end{pmatrix}, \quad x^\nu = \begin{pmatrix} 0 & 0 & 0 & 0 & 3 & -\frac{8}{25} & -\frac{3}{50} & 1 \end{pmatrix}^T,$$

$$e_{5\bullet}^\nu = \begin{pmatrix} \frac{1825}{2} & -100 & 0 & -3 \end{pmatrix}, \quad e_{6\bullet}^\nu = \begin{pmatrix} 104 & 12 & -8 & \frac{8}{25} \end{pmatrix},$$

$$e_{7\bullet}^\nu = \begin{pmatrix} -\frac{35}{2} & 2 & -1 & \frac{3}{50} \end{pmatrix}, \quad e_8^\nu = \begin{pmatrix} 0 & 0 & 0 & -1 \end{pmatrix}.$$

Step 1 (calculation x^1). Calculate the number q of the leading constraint equation using the formula:

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

Calculate the number r of the leading column using the formula:

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

Rewrite the constraint equation of the variable $x_6 \geqslant 0$ $\left(x_6^0 < 0\right)$:

$$x_2 = x_2^1 + e_{2\bullet}^1 \begin{pmatrix} x_1 & x_6 & x_3 & x_4 \end{pmatrix}^T,$$

$$x_2^1 = \frac{2}{75}, \quad e_{2\bullet}^1 = \begin{pmatrix} -\frac{26}{3} & \frac{1}{12} & \frac{2}{3} & -\frac{2}{75} \end{pmatrix},$$

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

Recalculate the elements of the constraint equations:

$$x_5 = x_5^1 + e_{5\bullet}^1 \begin{pmatrix} x_1 & x_6 & x_3 & x_4 \end{pmatrix}^T,$$

$$x_5^1 = \frac{1}{3}, \quad e_{5\bullet}^1 = e_{5\bullet}^0 E_2^1 = \begin{pmatrix} -\frac{275}{6} & \frac{25}{3} & -\frac{200}{3} & -\frac{1}{3} \end{pmatrix},$$

$$x_7 = x_7^1 + e_{7\bullet}^1 \begin{pmatrix} x_1 & x_6 & x_3 & x_4 \end{pmatrix}^T,$$

$$x_7^1 = -\frac{1}{150}, \quad e_{7\bullet}^1 = e_{7\bullet}^0 E_2^1 = \begin{pmatrix} -\frac{209}{6} & \frac{1}{6} & \frac{1}{3} & \frac{1}{150} \end{pmatrix},$$

$$x_8 = x_8^1 + e_{8\bullet}^1 \begin{pmatrix} x_1 & x_6 & x_3 & x_4 \end{pmatrix}^T,$$

$$x_8^1 = 1, \quad e_{8\bullet}^1 = e_{8\bullet}^0 E_2^1 = \begin{pmatrix} 0 & 0 & 0 & -1 \end{pmatrix}.$$

Recalculate the elements of the objective function:

$$e_{0\bullet}^1 = e_{0\bullet}^0 E_2^1 = \begin{pmatrix} \frac{3175}{3} & \frac{25}{3} & \frac{500}{3} & \frac{16}{3} \end{pmatrix}.$$

Set:

$$x^1 = \begin{pmatrix} 0 & \frac{2}{75} & 0 & 0 & \frac{1}{3} & 0 & -\frac{1}{150} & 1 \end{pmatrix}^T.$$

Step 2 (calculation x^2). Calculate the number q of the leading constraint equation using the formula:

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

Calculate the number r of the leading column using the formula:

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

Rewrite the constraint equation of the variable $x_7 \geqslant 0$ $\left(x_7^1 < 0\right)$:

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

Recalculate the elements of the constraint equations:

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

Recalculate the elements of the objective function:

$$e_{0\bullet}^2 = e_{0\bullet}^1 E_6^2 = \begin{pmatrix} 2800 & 50 & 150 & 5 \end{pmatrix}.$$

Set:

$$x^* = x^2 = \begin{pmatrix} 0 & \frac{3}{100} & 0 & 0 & 0 & \frac{1}{25} & 0 & 1 \end{pmatrix}^T \geqslant 0, \quad c^T x^* = -\frac{16}{3}.$$

EXAMPLE 3 (KLEE – SCHRIJVER [YAN, 2020]). Schrijver proved that the simplex method for solving the problem:

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

with Dantzig's rule is not polynomial: the number of iterations of the algorithm grows exponentially fast [Схрейвер, 1991]. The feasible domain of this problem is a deformed n-dimensional cube, so the simplex method sequentially passes through all its 2^n vertices. The reason for this is the combinatorial nature of the simplex method, which sequentially enumerates the vertices of the polyhedron of feasible solutions when searching for the optimal solution. Thus, this method is effective on a small set of input data, but as it increases, the complexity of the algorithm will increase abruptly.

Remark. $x^* = \begin{pmatrix} 0 & \cdots & 0 & 5^n \end{pmatrix}^T$ is the only optimal solution to the Klee – Schrijver problem [Схрейвер, 1991].

Let us show that the direct algorithm with Dantzig's rule finds a solution to the Klee – Schrijver problem at the first step of the first stage. The following auxiliary quantities are involved in the solution process:

c_0 $\left(-c_0 \leqslant cx^*\right)$ is an arbitrarily large scalar quantity,

x_0 $\left(x_0 \geqslant 0\right)$ is an additional variable,

$c_0 - x_0 + cx = 0$ is the constraint equation of an additional variable,

$\min\limits_{x \in R^n}(x_0)$, $c_0 - x_0 + cx = 0$, $h + Hx \geqslant 0$, $x_0 \geqslant 0$, $x \geqslant 0$, is a problem equivalent to the Klee – Schrijver problem,

$x_{n+1} \geqslant 0, \ldots, x_{2n} \geqslant 0$ are additional variables to reduce the Klee – Schrijver problem to canonical form by adding to the right-hand side of each equation, respectively.

Step 0 (initialization). Calculate the number r of the leading column (the number of the leading row $q = 0$) using the formula:

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

Rewrite the constraint equation of an additional variable $x_0 \geqslant 0$:

$$x_1 = x_1^0 + e^0_{1_0\bullet}\begin{pmatrix} x_0 & x_2 & \cdots & x_n \end{pmatrix}^T,$$

$$x_1^0 = 2^{-n+1}c_0, \quad e^0_{1_0\bullet} = \begin{pmatrix} -2^{-n+1} & -2^{-1} & \cdots & -2^{-n+2} & -2^{-n+1} \end{pmatrix},$$

$$E^0_{1_0} = \begin{pmatrix} -2^{-n+1} & -2^{-1} & \cdots & -2^{-n+2} & -2^{-n+1} \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}.$$

Set:

$$x^0 = \begin{pmatrix} 2^{-n+1}c_0 & 0 & \cdots & 0 \end{pmatrix}^T.$$

Step 1 (calculation x^1). Calculate the number q of the leading line using the formula:

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

Calculate the elements of the constraint equation of the nth row:

$$h_n^1 + \begin{pmatrix} h^1_{n1} & h^1_{n2} & \cdots & h^1_{nn} \end{pmatrix}\begin{pmatrix} x_0 & x_2 & \cdots & x_n \end{pmatrix}^T = x_{2n},$$

$$h_n^1 = h_n + h_{n1}x_1^0 = 5^n - 2c_0, \quad \begin{pmatrix} h^1_{n1} & h^1_{n2} & \cdots & h^1_{nn} \end{pmatrix} = h_n E^0_{1_0\bullet} = \begin{pmatrix} 2 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Calculate the number r of the leading column using the formula:

$$\Theta_r = \max_{i=1,\ldots,n,\ h^1_{ni}>0,\ i\neq 1} h^1_{ni} = h^1_{nn} = 1.$$

Rewrite the constraint equation of the nth row:

$$x_n = x_n^1 + e^1_{n_1\bullet}\begin{pmatrix} x_0 & x_2 & \cdots & x_{n-1} & x_{2n}\end{pmatrix}^T,$$
$$x_n^1 = -5^n + 2c_0, \quad e^1_{n_1\bullet} = \begin{pmatrix} -2 & 0 & \cdots & 0 & 1\end{pmatrix},$$
$$E^1_{n_1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ -2 & 0 & \cdots & 0 & 1\end{pmatrix}.$$

Recalculate the elements of the constraint equation:

$$x_1 = x_1^1 + e^0_{1_1\bullet}\begin{pmatrix} x_0 & x_2 & \cdots & x_{n-1} & x_{2n}\end{pmatrix}^T,$$
$$x_1^1 = -2^{-n+1}c_0 + 2^{-n+1}5^n, \quad e^0_{1_1\bullet} = e^0_{1_0\bullet}E^1_{n_1} = \begin{pmatrix} 2^{-n+1} & -2^{-1} & \cdots & -2^{-n+2} & -2^{-n+1}\end{pmatrix}.$$

Rewrite the constraint equation of the variable $x_1 \geqslant 0$ $\left(x_1^1 < 0\right)$:

$$x_0 = x_0^1 + e^1_{0_1\bullet}\begin{pmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_{2n}\end{pmatrix}^T,$$
$$x_0^1 = -5^n + c_0, \quad e^1_{0_1\bullet} = \begin{pmatrix} 2^{n-1} & 2^{n-2} & \cdots & 2 & 1\end{pmatrix},$$
$$E^1_{0_1} = \begin{pmatrix} 2^{n-1} & 2^{n-2} & \cdots & 2 & 1 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1\end{pmatrix}.$$

Recalculate the elements of the constraint equation:

$$x_n = x_n^2 + e^1_{n_2\bullet}\begin{pmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_{2n}\end{pmatrix}^T,$$
$$x_n^2 = 5^n, \quad e^1_{n_2\bullet} = e^1_{n_1\bullet}E^1_{0_1} = \begin{pmatrix} -2^n & -2^{n-1} & \cdots & -2^2 & -1\end{pmatrix}.$$

Set:

$$x^* = x^1 = \begin{pmatrix} 0 & 0 & \cdots & 0 & 5^n\end{pmatrix}^T \geqslant 0, \quad cx^* = -5^n.$$

EXAMPLE 4 (SVIRIDENKO [СВИРИДЕНКО, 2017]). Large LP problems, as a rule, have a nonunique solution. Various methods for solving such problems (simplex method, interior point method, quadratic penalty function method) lead to different solutions in the case of nonuniqueness. The simplex method provides a solution that belongs to the vertex of a polyhedral set. Interior point methods converge to a solution in which the condition of strict complementary nonrigidity is satisfied. The external quadratic penalty function method makes it possible to find an exact normal solution (in other words, a solution with a minimum Euclidean norm). Note that finding a normal solution is closely related to the regularization method [Поляк, 2006] and the quadratic penalty function method [Фиакко, Мак-Кормик, 1972].

An approach to constructing all the vertices of $\left\{x^{1^*}, \ldots, x^{s^*}\right\}$ of a polyhedral set of solutions to an LP problem using a direct algorithm is considered using a problem of type (1) with an infinite set of solutions [Свириденко, 2017]:

$$c = \begin{pmatrix} 1 & 1 & 2 & 2 & 1 \end{pmatrix},$$
$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & -1 & 2 & 2 & 3 \end{pmatrix}, \quad h = \begin{pmatrix} 2 \\ 2 \end{pmatrix},$$
$$x^{1^*} = \begin{pmatrix} 0 & \frac{2}{3} & \frac{4}{3} & 0 & 0 \end{pmatrix}^T, \quad cx^{1^*} = \frac{10}{3},$$
$$x^{2^*} = \begin{pmatrix} 0 & \frac{2}{3} & 0 & \frac{4}{3} & 0 \end{pmatrix}^T, \quad cx^{2^*} = \frac{10}{3},$$
$$x^{3^*} = \begin{pmatrix} 0 & \frac{6}{3} & 0 & 0 & \frac{4}{3} \end{pmatrix}^T, \quad cx^{3^*} = \frac{10}{3}.$$

Now consider the approach to calculating the normal solution x^* as a solution to the problem of the projection of zero onto the set of solutions to the Sviridenko problem:

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

The solution to this problem exists and is unique [Малозёмов, 2017b]. Its solution is

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

and hence

$$x^* = \frac{1}{2}x^{1^*} + \frac{1}{2}x^{2^*} + 0x^{3^*} = \begin{pmatrix} 0 & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} & 0 \end{pmatrix}^T, \quad cx^* = \frac{10}{3}.$$

Conclusion

The direct method is a sequence of steps, at each of which zeros are obtained in the required positions of the next processed column of the matrix of the conditions of the direct problem. In this case, the zeros obtained previously in preceding columns are preserved.

Lachinov and Polyakov in [Лачинов, Поляков, 1999] noted that any modern programming language has the "taste" and "smell" of its predecessors, and so does the direct algorithm, which inevitably bears the features of the simplex method with an inverse matrix. Moreover, two variants of the algorithm for solving the problem obtained at the first stage in conjugate canonical form are proposed. The first is based on solving it using a direct algorithm in terms of the simplex method, and the second is based on solving the problem dual to it using the simplex method. It has been proven that both options are equivalent: with the same initial data, they generate the same sequence of points — the basic solution and the current dual solution of the vector of row estimates. Hence, the following conclusion follows: a direct algorithm is an algorithm similar to the simplex method.

At each iteration of each stage, the direct algorithm generates a sequence of points that extend beyond the feasible region in the direction of increasing the objective function, until a feasible solution is obtained, and the hyperplane of the objective function passing through these points does not intersect the feasible region. Therefore, in accordance with the terminology of Fiacco and McCormick, the direct algorithm is one of the variants of the external point method [Фиакко, Мак-Кормик, 1972].

The simplex method, the interior point method and the quadratic penalty function method lead to different solutions if they are not unique . The simplex method provides a solution that belongs to the vertex of a polyhedral set. Interior point methods converge to a solution in which the condition of strict complementary nonrigidity is satisfied . The external quadratic penalty function method makes it possible to find a solution with a minimum Euclidean norm (exact normal solution). By solving the problem from Example 4 in § 4 (an approach to calculating a normal solution), we will show that the direct method makes it possible to find all the optimal vertices $\left(x^{1^*}, x^{2^*}, x^{3^*}\right)$ of a polyhedral set and, therefore, leads to both an exact normal solution and a solution in which the complementary slackness condition is satisfied. The solution $\left(x^{1^*}\right)$, belonging to the vertex of the polyhedral set, is achieved at the second iteration of the first stage:

$$\min(x_1), \quad x_2 = \frac{2}{3} + \frac{1}{3}x_1 + x_5, \quad x_3 = \frac{4}{3} - \frac{1}{3}x_1 - x_4 - x_5, \quad x_i \geqslant 0, \quad i = 1, \ldots, 5,$$
$$x^{1^*} = \begin{pmatrix} 0 & \frac{2}{3} & \frac{4}{3} & 0 & 0 \end{pmatrix}^T, \quad cx^{1^*} = \frac{10}{3}.$$

To find the remaining solutions $\left(x^{2^*}, x^{3^*}\right)$, belonging to the vertices of the polyhedral set, we solve LP problems (the optimal value x_1 is obviously zero):

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

Their solutions are

$$\begin{pmatrix} x_2 & x_3 & x_4 & x_5 \end{pmatrix}^T = \begin{pmatrix} \frac{2}{3} & 0 & \frac{4}{3} & 0 \end{pmatrix}^T, \quad \begin{pmatrix} x_2 & x_3 & x_4 & x_5 \end{pmatrix}^T = \begin{pmatrix} 0 & \frac{6}{3} & 0 & \frac{4}{3} \end{pmatrix}^T,$$

hence,

$$x^{2^*} = \begin{pmatrix} 0 & \frac{2}{3} & 0 & \frac{4}{3} & 0 \end{pmatrix}^T, \quad cx^{2^*} = \frac{10}{3}, \quad x^{3^*} = \begin{pmatrix} 0 & \frac{6}{3} & 0 & 0 & \frac{4}{3} \end{pmatrix}^T, \quad cx^{3^*} = \frac{10}{3}.$$

Acknowledgments

The author expresses gratitude to the reviewer for his impressive contribution to the writing of the work.

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 razrezhennyj 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 putey 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, 2017a. — 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, 2017b. — 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., Иванилов 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 nulja 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. Prjamye 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. Prjamye 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 Wickunde 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).