ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2018, Vol. 58, No. 2, pp. 159–169. © Pleiades Publishing, Ltd., 2018.
Original Russian Text © B.V. Ganin, A.I. Golikov, Yu.G. Evtushenko, 2018, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2018, Vol. 58, No. 2, pp. 169–180.

Dedicated to the 100th birthday of Academician N.N. Moiseev

Projective-Dual Method for Solving Systems of Linear Equations with Nonnegative Variables

B. V. Ganin, A. I. Golikov*, and Yu. G. Evtushenko

Dorodnitsyn Computing Center, FRC CSC RAS, Moscow, 119333 Russia

**e-mail gol-a@yandex.ru*

Received July 12, 2017; in final form, September 6, 2017

Abstract—In order to solve an underdetermined system of linear equations with nonnegative variables, the projection of a given point onto its solutions set is sought. The dual of this problem—the problem of unconstrained maximization of a piecewise-quadratic function—is solved by Newton's method. The problem of unconstrained optimization dual of the regularized problem of finding the projection onto the solution set of the system is considered. A connection of duality theory and Newton's method with some known algorithms of projecting onto a standard simplex is shown. On the example of taking into account the specifics of the constraints of the transport linear programming problem, the possibility to increase the efficiency of calculating the generalized Hessian matrix is demonstrated. Some examples of numerical calculations using MATLAB are presented.

Keywords: systems of linear equations with nonnegative variables, regularization, projection of a point, duality, generalized Newton's method, unconstrained optimization, transport linear programming problem.

DOI: 10.1134/S0965542518020057

INTRODUCTION

The problem of finding a solution of an underdetermined system of linear equations with nonnegative variables does not belong to the classical problems of computational linear algebra. To solve such problems, it is expedient to use duality theory and various optimization methods, e.g., Newton's method, which, being applied to large linear programming (LP) problems, finds the projection of a given point onto the solution set of the initial LP problem [1, 2].

On the one hand, a system of linear equations with nonnegative variables can be considered as an LP problem with a zero objective function. On the other hand, as is well known, an LP problem reduces to solving a system of linear equations and inequalities. The disadvantage of this reduction is an increased number of variables in the system in comparison with the initial LP problem. A different method of reducing the initial primal LP problem to a system of linear equations with nonnegative variables was proposed in [1]. If, in the primal problem, we add to equality constraints with nonnegative variables the equality of the objective function to its a priori unknown optimal value, we will obtain a linear system describing the solution set of the primal LP problem. Let us find in this set the projection of a given point, i.e., the solution of a strictly convex quadratic programming problem (the projection stage). To solve this problem, it is expedient to pass to the dual problem, which is the problem of *unconstrained* maximization of a concave piecewise-quadratic function (the second stage of the method is the passage to the dual problem). The number of variables in the dual problem is equal to the number of constraints in the primal LP problem plus the Lagrange multiplier corresponding to an additional constraint: the equality of the objective LP function to its unknown optimal value. As shown in [1], this factor can be fixed, taking it larger than a certain threshold value. After this, the knowledge of the optimal value of the objective function is not required and, based on the results of unconditional maximization, the projection of a given point on the solution set of the LP problem is found by a simple formula.

The *unconstrained* maximization of a concave piecewise-quadratic function can be carried out especially efficiently by Newton's method, which, in this case, converges globally for a finite number of steps

if the step size is chosen by the Armijo rule [3]. Since a piecewise-quadratic function does not have a matrix of second derivatives, a generalized Hessian matrix is used. In some cases, this matrix has a specific form that makes it possible to recalculate it fairly easily at each step of Newton's method and makes it possible to easily parallelize the method [2, 4, 5].

In Section 1, the solution of a linear system of equations with nonnegative variables is considered as a problem of projecting a point on the solution set of this system (the first stage of the method is the replacement of the original problem by the problem of finding the projection of a given point on its solution set). Thus, we come to a convex quadratic programming problem, the dual of which is the problem of *unconstrained* maximization of a concave piecewise-quadratic function [6]. From the solution of the dual problem, the projection, i.e., the solution of the original system, is calculated by a simple formula (the second stage of the method is the passage to the dual problem and its solution). The number of variables in the dual problem is equal to the number of equations in the original system.

In Section 2, the linear system with nonnegative variables is solved as a regularization problem, which is the problem of minimizing a strictly convex quadratic function on a nonnegative orthant. For this minimization problem, the dual problem of *unconstrained* maximization of a strictly concave piecewise-quadratic function is introduced. From its solution, the solution of the regularized problem is easily calculated. In contrast to the primal minimization problem on a nonnegative orthant, this dual problem of *unconstrained* maximization can be efficiently solved by Newton's method. Examples of solving large linear systems in MATLAB by the projection and regularization method are presented.

In Section 3, we consider the special problem of projecting a point onto a standard simplex. Using the duality and Newton's method, we show the connection between the methods of projecting onto a simplex, which were considered in detail in [9, 10].

In Section 4, in order to reduce the computational costs in the construction of the generalized Hessian matrix, we consider and take into account the specificity of linear equality constraints in the transport linear programming problem. The results of calculations in MATLAB are presented.

1. BASIC COMPUTATIONAL FORMULAS FOR THE PROJECTION-DUAL AND NEWTON'S METHODS

Given a system of linear equations with nonnegative variables:

$$Ax = b, \quad x \geq 0_n. \tag{1}$$

Here, the matrix $A \in \mathbb{R}^{m\times n}$ and the vector $b \in \mathbb{R}^m$ are defined and $m \leq n$. We denote by 0_i the i-dimensional zero vector. Henceforward, we will assume that system (1) is solvable. This system is regarded as a special case of the linear programming problem

$$\min_{x\in X} 0_n^\top x, \quad X = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0_n\}, \tag{2}$$

with the objective function vector $c = 0_n$. The problem dual to (2) is

$$\max_{u\in U} b^\top u, \quad U = \{u \in \mathbb{R}^m : A^\top u \leq 0_n\}. \tag{3}$$

Since $X \neq \emptyset$ and $U \neq \emptyset$, LP problems (2) and (3) are solvable. Denote their solution sets by X^* and U^*. In this case, $X = X^*$, i.e., the feasible set of problem (2) coincides with its solution set. The specific form of these linear programming problems allows us to use, for solving problem (2) and, consequently, (1), a concave piecewise-quadratic penalty function not containing a penalty coefficient. In order to find the normal solution of system (1), it suffices to apply to dual problem (3) a quadratic penalty function. Then, from the solution u^* of the problem

$$\max_{u\in\mathbb{R}^m}\left\{b^\top u - \frac{1}{2}\left\|(A^\top u)_+\right\|^2\right\} \tag{4}$$

we find by a simple formula $x^* = (A^\top u^*)_+$: the normal solution of LP problem (2) (the normal solution of system (1)) [1]. Hereinafter, a_+ denotes a vector in which all negative components are replaced by zeros.

The normal solution of system (1) (LP problem (2)) is the projection of the zero point onto the set X (X^*): $x^* = \Pr_X(0_n) = \Pr_{X^*}(0_n)$. Note that problem (4) is dual to the quadratic programming problem (see [1, 6])

$$\min_{x\in X}\frac{1}{2}\|x\|^2, \quad X = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0_n\},$$

or, which is the same, to regularized problem (2).

In a similar manner, we find the projection $\hat{x}^*$ of a given point $\hat{x}$ onto the set X. The following two problems are mutually dual [1, 6]:

$$\min_{x\in X}\frac{1}{2}\|x - \hat{x}\|^2, \quad X = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0_n\}, \tag{5}$$

$$\max_{p\in\mathbb{R}^m}\left\{b^\top p - \frac{1}{2}\left\|(\hat{x} + A^\top p)_+\right\|^2 + 1/2\|\hat{x}\|^2\right\}. \tag{6}$$

By the Frank–Wolfe theorem [7], problem (5) is solvable, since its objective function is quadratic and bounded below on the nonempty set X. Therefore, dual problem (6) is also solvable [6].

From duality theory for quadratic programming, we deduce the following assertion.

Assertion [6]. *The unique solution $\hat{x}^*$ of problem* (5) *is expressed in terms of the solution p^* of dual problem* (6) *by the formula*

$$\hat{x}^* = (\hat{x} + A^\top p^*)_+. \tag{7}$$

Thus, solving the unconstrained minimization problem

$$\max_{p\in\mathbb{R}^m} S(p), \quad S(p) = b^\top p - \frac{1}{2}\left\|(\hat{x} + A^\top p)_+\right\|^2, \tag{8}$$

we find by formula (7) the solution $\hat{x}^*$ of the linear system with nonnegative variables (1), which is the projection of the given point $\hat{x}^* = \Pr_X(\hat{x})$ onto the set X.

The function $S(p)$ does not have the ordinary Hessian matrix. Indeed, the gradient

$$S_p(p) = b - A(\hat{x} + A^\top p)_+$$

of the functions $S(p)$ is not differentiable. However, for this function, we can define a generalized Hessian matrix, which is a symmetrically negative-semidefinite $m \times m$ matrix of the form

$$S_{pp}(p) = -AD(z)A^\top, \tag{9}$$

where $D(z)$ denotes a $n \times n$ diagonal matrix with the ith diagonal element z_i equal to 1 if $(\hat{x} + A^\top p)_i > 0$ and equal to 0 if $(\hat{x} + A^\top p)_i \leq 0$, $i = 1,\ldots,n$; i.e., the n-dimensional vector $z = \operatorname{sgn}(\hat{x} + A^\top p)_+$.

Since the generalized Hessian matrix $S_{pp}(p^k)$ at the kth step of Newton's method can be degenerate, we form the matrix $H^k \in \mathbb{R}^{m\times m}$:

$$H^k = -AD(z^k)A^\top - \delta I_m, \tag{10}$$

where $I_m \in \mathbb{R}^{m\times m}$ is the unit matrix and δ is some positive number (usually, 10^{-4}). The direction of maximization, Δp, of the function $S(p)$ at the point p^k is found from the solution of the system of linear equations

$$H^k\Delta p = -S_p(p^k). \tag{11}$$

The next approximation p^{k+1} is defined by the formula $p^{k+1} = p^k - \tau^k\Delta p$, where the step τ^k is found from the solution of the one-dimensional maximization problem $\tau^k = \arg\max_\tau S(p^k - \tau\Delta p)$, e.g., by the Armijo method. O. Mangasarian proved the finite convergence from an arbitrary point p^0 of the generalized Newton's method for the unconstrained maximization of a concave piecewise-quadratic function [3]. Thus, in a finite number of steps, we find the solution p^* of problem (6) and find by formula (7) the pro-

jection of the point $\hat{x}$ onto the set X. Thus, the solution of system (1) with n nonnegative variables and m equations reduces to a finite number of solutions of systems (11) of the dimension $m \times m$.

2. REGULARIZED PROBLEM OF FINDING THE PROJECTION OF A POINT

In order to find the projection of a point $\hat{x}$ onto the solution set of system (1), we can use another approach, different from the application of mutually dual problems (5) and (6). For this, following [8], let us consider the regularized problem (or, which is the same, penalty problem (5))

$$\min_{x \in \mathbb{R}^n_+} F(x), \quad F(x) = \frac{1}{2}(\|b - Ax\|^2 + \varepsilon \|x - \hat{x}\|^2) \tag{12}$$

with a positive parameter ε. As $\varepsilon \to 0$, the solution $x(\varepsilon)$ of problem (12) converges to the solution $\hat{x}^*$ of problem (5) or, which is the same, to the projection $\hat{x}^*$ of the point $\hat{x}$ onto the solution set of system (1). From the computational point of view, this problem has two drawbacks: the condition of nonnegativity of variables x and their dimension $n \gg m$, which hampers the direct application of Newton's method to problem (12).

These drawbacks are not inherent in the following unconstrained maximization problem:

$$\max_{u \in \mathbb{R}^m} W(u), \quad W(u) = b^\top u - \frac{\varepsilon}{2}\left\|(\hat{x} + \frac{1}{\varepsilon}A^\top u)_+\right\|^2 - \frac{1}{2}\|u\|^2, \tag{13}$$

the solution of which can be easily calculated in terms of the solution of problem (12).

Theorem. *For any $\varepsilon > 0$, the unique solution $x(\varepsilon) = \arg\min_{x \in \mathbb{R}^n_+} F(x)$ of problem* (12) *is determined in terms of the unique solution $u(\varepsilon) = \arg\max_{u \in \mathbb{R}^m} W(u)$ of problem* (13) *by the formula*

$$x(\varepsilon) = \left(\hat{x} + \frac{1}{\varepsilon}A^\top u(\varepsilon)\right)_+.$$

Proof. The function to be minimized in problem (12) for $\varepsilon > 0$ is a strictly convex quadratic function, bounded below by zero on $\mathbb{R}^n_+$. Therefore, by the Frank–Wolfe theorem [7], problem (12) always has a solution, which is unique.

In problem (13), the maximized piecewise-quadratic function for $\varepsilon > 0$ is strictly concave and bounded from above on the entire space $\mathbb{R}^m$. Indeed, it is true that

$$W(u) = b^\top u - \frac{\varepsilon}{2}\left\|(\hat{x} + \frac{1}{\varepsilon}A^\top u)_+\right\|^2 - \frac{1}{2}\|u\|^2 = \frac{1}{2}\|b\|^2 - \frac{1}{2}\|b\|^2 + b^\top u - \frac{1}{2}\|u\|^2$$
$$- \frac{\varepsilon}{2}\left\|(\hat{x} + \frac{1}{\varepsilon}A^\top u)_+\right\|^2 = \frac{1}{2}\|b\|^2 - \frac{1}{2}\|b - u\|^2 - \frac{\varepsilon}{2}\left\|(\hat{x} + \frac{1}{\varepsilon}A^\top u)_+\right\|^2 \le \frac{1}{2}\|b\|^2.$$

Therefore, for any $\varepsilon > 0$, by the Frank–Wolfe theorem, problem (13) always has a solution, which is unique.

For problem (12), using additional variables $y \in \mathbb{R}^m$, we introduce artificial constraints $Ax + y = b$ and obtain an equivalent nonlinear programming problem:

$$\min_{y \in \mathbb{R}^m} \min_{x \in \mathbb{R}^n_+} \frac{1}{2}(\|y\|^2 + \varepsilon \|x - \hat{x}\|^2), \quad Ax + y = b, \tag{14}$$

for which the dual problem is formulated in a standard way.

Introduce the Lagrange function for problem (14) as follows:

$$L(y, x, u) = \frac{1}{2}\|y\|^2 + \frac{\varepsilon}{2}\|x - \hat{x}\|^2 + u^\top(b - Ax - y).$$

Here, $u \in \mathbb{R}^m$ are the Lagrange multipliers for this problem. The problem dual to (14) has the form

$$\max_{u \in \mathbb{R}^m} \min_{y \in \mathbb{R}^m} \min_{x \in \mathbb{R}^n_+} L(y, x, u). \tag{15}$$

Write the conditions for the minimum with respect to y and x for the internal minimization problem:

$$L_y(y(\varepsilon), x(\varepsilon), u) = y(\varepsilon) - u = 0_m,$$
$$L_x(y(\varepsilon), x(\varepsilon), u) = \varepsilon(x(\varepsilon) - \hat{x}) - A^\top u \geq 0_n, \quad x^\top(\varepsilon)(\varepsilon(x(\varepsilon) - \hat{x}) - A^\top u) = 0, \quad x(\varepsilon) \geq 0_n.$$

From these conditions, we easily find the solutions of internal minimization problem (15):

$$y(\varepsilon) = u, \tag{16}$$

$$x(\varepsilon) = \left(\hat{x} + \frac{1}{\varepsilon}A^\top u\right)_+. \tag{17}$$

Substituting solutions (16) and (17) into the Lagrange function $L(y, x, u)$, after simple transformations, we obtain the dual function for problem (15):

$$\tilde{L}(u) = b^\top u - \frac{\varepsilon}{2}\left\|(\hat{x} + \frac{1}{\varepsilon}A^\top u)_+\right\|^2 - \frac{1}{2}\|u\|^2 + \frac{\varepsilon}{2}\|\hat{x}\|^2 = W(u) + \frac{\varepsilon}{2}\|\hat{x}\|^2,$$

i.e., we come to problem (13) dual to problem (14) and, therefore, to problem (12). The theorem is proven.

Let us make in (13) the change of variables $p = u/\varepsilon$ and solve instead of (13) the problem

$$\max_{p\in\mathbb{R}^m} W(p), \quad W(p) = b^\top p - \frac{1}{2}\left\|(\hat{x} + A^\top p)_+\right\|^2 - \frac{\varepsilon}{2}\|p\|^2. \tag{18}$$

Then, from its solution $p(\varepsilon)$, we find the solution of regularized problem (12):

$$x(\varepsilon) = (\hat{x} + A^\top p(\varepsilon))_+.$$

Problem (18) can be conveniently solved by Newton's method, especially if we take into account that $m \ll n$. The function $W(p)$ to be maximized in problem (18) is strictly concave piecewise-quadratic and differentiable. Its gradient has the form

$$W_p(p) = b - A(\hat{x} + A^\top p)_+ - \varepsilon p.$$

The function $W_p(p)$ is not differentiable; we can define for it a generalized Hessian matrix, which, in contrast to the matrix S_{pp}, for $\varepsilon > 0$, is a nondegenerate $(m \times m)$-matrix of the following form:

$$W_{pp}(p) = -AD(z)A^\top - \varepsilon I_m,$$

Here, by analogy with formula (9), we denote by $D(z)$ the diagonal $(n \times n)$-matrix, and the n-dimensional vector is $z = \operatorname{sgn}(\hat{x} + A^\top p)_+$.

Let us present some results of calculations in MATLAB 2014b on a personal computer with an AMD FX-8350 processor with a frequency of 3.90 GHz and 16 GB RAM. We solved randomly generated systems with a large number of nonnegative variables (up to a million) and a moderate number of equations (up to four thousand), i.e., the condition $n \gg m$ was satisfied. The numbers m and n, which determine the number of rows and columns in the matrix A, and d, the density of filling the matrix A with nonzero elements, were specified. In particular, $d = 1$ means that all elements of the matrix A were randomly generated and $d = 0.1$ indicates that only 10% of the elements of the matrix A were generated and the rest were set to zero. The elements of the matrix A were specified randomly from the interval $[-50, +50]$. The solution p^* was chosen randomly from the interval $[-10, 10]$, and the projected vector $\hat{x}$, from $[0, 10]$. Using them, by formula (7), the projection $\hat{x}^*$ and then the vector $b = A\hat{x}^*$ were calculated.

Table 1 presents the dimensions of the problems and the density of the matrix. The number of iterations of Newton's method, the Chebyshev norm of the residual of the system, and the computation time are given for the solution of projection problem (5) by solving dual problem (8) by Newton's method and for the solution of regularized problem (12) by solving the dual problem (18) by Newton's method. In regularized problem (12), the regularization coefficient ε was set to 10^{-10} and the unconstrained maximization problem was solved. When solving dual problem (8), at each kth step of Newton's method, in formula (10) for calculating H^k, it was assumed that $\delta = 10^{-4}$. In these examples, the use of the Armijo rule was not necessary. From the calculation results, it can be seen that projection problem (5) and regularized problem (12) showed approximately the same accuracy and solution time.

Table 1. Comparative results of programs for solving problems (5) and (12)

Dimension	Density	Problem (5)			Problem (12)		
$m \times n$	d	it	$\|Ax - b\|_\infty$	t, s	it	$\|Ax - b\|_\infty$	t, s
250×10^4	1	6	2.8×10^{-11}	2.8	6	2.6×10^{-11}	2.7
500×10^4	1	6	5.6×10^{-11}	8.7	6	3.7×10^{-11}	8.9
1000×10^4	1	7	6.0×10^{-11}	35.5	7	4.1×10^{-11}	35.1
2000×10^4	1	8	4.4×10^{-11}	166.4	8	3.7×10^{-11}	156.1
3000×10^4	1	9	6.2×10^{-11}	446.1	9	6.5×10^{-11}	451.6
4000×10^4	1	10	7.4×10^{-11}	822.1	10	7.0×10^{-11}	825.1
1000×10^5	0.5	6	1.8×10^{-10}	167.6	6	1.4×10^{-10}	164.9
2000×10^5	0.5	6	5.3×10^{-10}	671.7	6	5.0×10^{-10}	658.9
3000×10^5	0.5	6	4.5×10^{-10}	1479.0	6	5.3×10^{-10}	1414.0
4000×10^5	0.5	6	5.2×10^{-10}	2508.5	6	5.9×10^{-10}	2470.3
500×10^6	0.1	6	1.5×10^{-10}	57.4	6	2.2×10^{-10}	59.3
1000×10^6	0.1	6	1.5×10^{-9}	180.5	6	1.6×10^{-9}	180.1
2000×10^6	0.1	6	1.3×10^{-9}	685.8	6	1.2×10^{-9}	687.6

3. SPECIAL CASE OF PROJECTING A POINT ONTO A STANDARD SIMPLEX

The projective-dual method is especially simple and efficient in a special case of problem (5) when the projection of a given point $\hat{x} \in \mathbb{R}^n$ on a standard simplex is sought. Suppose that, in problem (5), $m = 1$, $b = 1$, and $A = e_n^\top$, where e_n denotes an n-dimensional vector whose all components are equal to unity, i.e., a standard simplex is defined. Write this projection problem in the form

$$\min_{x \in X} \frac{1}{2} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2, \quad X = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0,\ i \in N = 1:n \right\}. \tag{19}$$

In accordance with (6), the problem dual to (19) reduces to the unconstrained maximization of a concave one-dimensional piecewise-quadratic function of the scalar $p \in \mathbb{R}^1$:

$$\max_{p \in \mathbb{R}^1} S(p) = \max_{p \in \mathbb{R}^1} \left\{ p - \frac{1}{2} \left\| (\hat{x} + e_n p)_+ \right\|^2 \right\} = \max_{p \in \mathbb{R}^1} \left\{ p - \frac{1}{2} \sum_{i=1}^{n} (\hat{x}_i + p)_+^2 \right\}. \tag{20}$$

Substituting the solution p^* of this problem into formula (7), we find the projection $\hat{x}^*$ of the point $\hat{x}$ onto the standard simplex:

$$\hat{x}^* = (\hat{x} + e_n p^*)_+. \tag{21}$$

The one-dimensional unconstrained maximization problem (19) can be solved using various optimization algorithms, but Newton's method is especially efficient. Let us show that the "vector algorithm" studied in detail in [9, 10] is a variant of Newton's method with a special choice of the initial approximation p^0, which ensures the positivity of the generalized second derivative of the function $S(p)$ at each step.

The derivative of the function $S(p)$, calculated at the kth iteration of the generalized Newton's method at the point p^k, has the form

$$S_p(p) = 1 - e_n^\top (\hat{x} + e_n p^k)_+ = 1 - \sum_{i=1}^{n} (\hat{x}_i + p^k)_+ = 1 - \sum_{i \in N \setminus I_k} \hat{x}_i + n_k p^k.$$

Here, we introduce the index set $I_k = \{ i \in N : \hat{x}^i + p^k \leq 0 \}$ and $n_k = |N \setminus I_k|$ is the number of positive components of the vector $\hat{x}^k = (\hat{x} + e_n p^k)_+$.

The equality to zero of the derivative $S_p(p)$ is the necessary and sufficient condition for the optimality of problem (20). It should be noted that finding the root p^* of the equation

$$\sum_{i=1}^{n}(\hat{x}_i + p)_+ = 1 \tag{22}$$

underlies the "scalar algorithm" presented in [9, 10].The high efficiency of this algorithm is related to the specificity of the equation, whose left-hand side is a piecewise-linear function of one variable p. Basically, solving Eq. (22) by any method, we obtain by the formula (21) the solution of the original problem (19).

The generalized second derivative $S_{pp}(p)$ of a function $S(p)$, calculated at a point p^k has the form

$$S_{pp}(p^k) = -e_n^\top D(z^k)e_n = -n_k. \tag{23}$$

Here, $D(z^k)$ is a $n \times n$ diagonal matrix in which the ith diagonal element is equal to the ith component of the vector z^k, which is equal to 1 if $(\hat{x} + e_n p^k)_i > 0$ and equal to 0 if $(\hat{x} + e_n p^k)_i \le 0$, $i = 1,\ldots,n$, i.e., the n-dimensional vector $z^k = \operatorname{sgn}(\hat{x} + e_n p^k)_+$.

An iteration of Newton's method has the form

$$p^{k+1} = p^k - \frac{S_p(p^k)}{S_{pp}(p^k)} = \frac{1}{n_k}\left(1 - \sum_{i\in N\setminus I_k}\hat{x}_i\right). \tag{24}$$

The case of $n_k = 0$ is possible if $N\setminus I_k = \emptyset$, i.e., if all components of the vector $\hat{x} + e_n p^k$ are nonpositive. In this case, in accordance with (10), we should use in method (24) the regularized generalized second derivative

$$H^k = S_{pp}(p^k) - \delta = -n_k - \delta, \tag{25}$$

where δ is some positive number (usually, 10^{-4}).

Method (24) converges in a finite number of steps. The choice in the "vector algorithm" [9, 10] of the initial approximation by the formula

$$p^0 = \frac{1}{n}\left(1 - \sum_{i\in N}\hat{x}_i\right) \tag{26}$$

guarantees that the second derivative $S_{pp}(p)$ will be nonzero at each step. During the work of the algorithm, the index set I_k strictly expands and, at a finite iteration, is equal to the number of zero components in the projection $\hat{x}^*$, the solution of original problem (19). The total number of iterations depends on the initial approximation p^0 and the number of positive components in the solution $\hat{x}^*$.

Let us consider an example from [9] with the maximum complexity of the "vector algorithm," when the number of positive components of the vector $\hat{x}^k = (\hat{x} + e_n p^k)$ decreases by one at each iteration. In this case, the solution of the problem of projecting a point onto a simplex is achieved at one of the unit vectors of the space $\mathbb{R}^n$.

Example (see [9]). Let $n = 4$, $\hat{x} = (1,17,22,25)$, and the initial approximation $p^0 = -16$ be chosen according to (26). In this case, at each iteration, one component of the current vector x changes and the method arrives at the optimal point $\hat{x}^* = (0,0,0,1)$ in 4 iterations. However, for any initial approximation p^0 from the interval $(-25, -22)$, Newton's method (24) finds the projection in one step; for any p^0 from $(-22, -17)$, in two steps; for any p^0 from $(-17, -1)$, for three steps; and, for $p^0 > -1$, in four steps.

If we take $p^0 < -25$, then all components of the vector $(\hat{x} + e_4 p^0)$ are negative, we must use in Newton's method generalized derivative (25), and the projection is obtained at the sixth step.

4. THE CASE OF A TRANSPORT LP PROBLEM

Given a transport problem in the standard formulation:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min,$$
$$\sum_{j=1}^{n} x_{ij} = a_i, \quad \sum_{i=1}^{m} x_{ij} = b_j, \quad x_{ij} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n. \tag{T}$$

Define the vector $\overline{c} \in \mathbb{R}^{mn}$ as $\overline{c} = (c_{11}c_{12}...c_{1n}...c_{m1}c_{m2}...c_{mn})^{\top}$, the vector $\overline{b} \in \mathbb{R}^{m+n}$ as $\overline{b} = (a_1...a_m, b_1...b_n)^{\top}$, and the vector $\overline{x} \in \mathbb{R}^{mn}$ as $\overline{x} = (x_{11}x_{12}...x_{1n}...x_{m1}x_{m2}...x_{mn})^{\top}$. Then, the problem (T) can be rewritten in the form of a linear programming problem in the canonical form:

$$\overline{c}^{\top}\overline{x} \to \min, \quad \overline{A}\overline{x} = \overline{b}, \quad \overline{x} \geq 0. \tag{P}$$

Here, the matrix $\overline{A}$ of the dimension $(m+n)\times mn$ consists of zeros and units and has the form

$$\overline{A} = \begin{pmatrix} 1 & 1 & \dots & 1 & & & & & & & & \\ & & & & 1 & 1 & \dots & 1 & & & & \\ & & & & & & & & & & & \dots \\ 1 & & & & 1 & & & & 1 & & & \dots \\ & 1 & & & & 1 & & & & 1 & & \\ & & \ddots & & & & \ddots & & & & \ddots & \\ & & & 1 & & & & 1 & & & & 1 \end{pmatrix}.$$

As shown in [1], the linear programming problem (P) reduces to the following unconstrained maximization problem:

$$\max_{p\in\mathbb{R}^{(m+n)}} S(p,\beta,\hat{\overline{x}}) = \max_{p\in\mathbb{R}^{(m+n)}} \left\{\overline{b}^{\top}p - \frac{1}{2}\left\|(\hat{\overline{x}} + \overline{A}^{\top}p - \beta\overline{c})_{+}\right\|^2\right\}. \tag{27}$$

Here, β is some numerical parameter and $\hat{\overline{x}}$ is an arbitrary fixed point. Then, for any $\beta \geq \beta_*$, where β_* is a certain threshold value, one can find the projection of the point $\hat{\overline{x}}$ onto the solution set of the linear programming problem (P) by the formula

$$\hat{\overline{x}}^* = (\hat{\overline{x}} + \overline{A}^{\top}p(\beta) - \beta\overline{c})_{+},$$

where $p(\beta)$ is the solution of problem (27).

Problem (27) can be efficiently solved by the generalized Newton's method. Since the Hessian matrix for the concave piecewise-quadratic function $S(p, \beta, \hat{x})$ is not defined, we use the generalized Hessian matrix, which is a diagonal $(m+n)\times(m+n)$-matrix

$$S_{pp}(p,\beta,\hat{\overline{x}}) = -\overline{A}D(z)\overline{A}^{\top}. \tag{28}$$

Here, $D(z)$ is a diagonal $mn\times mn$-matrix whose ith diagonal element is the ith component of the vector $z \in \mathbb{R}^{mn}$, which is equal to 1 if $(\hat{\overline{x}} + \overline{A}^{\top}p - \beta\overline{c})_i > 0$ and equal to 0 in the remaining cases, $i = 1,...,mn$; i.e., $z = \operatorname{sgn}(\hat{\overline{x}} + \overline{A}^{\top}p - \beta\overline{c})_{+}$.

When using the generalized Newton's method, much time is spent to calculate matrix (28). Therefore, in order to accelerate the computation of generalized Hessian matrix (28), it is expedient to take into account the specific form of the constraint matrix $\overline{A}$ of the transport problem. The matrix $\overline{A}$ has a pronounced block structure, and the generalized Hessian matrix has the same property. In particular, if $D(z) = I_{mn}$, matrix (28) has the form

$$S_{pp} = \overline{A}\overline{A}^{\top} = \begin{pmatrix} \tilde{P} & \tilde{Q} \\ \tilde{Q}^T & \tilde{R} \end{pmatrix},$$
$$\tilde{P} = mI_n, \quad \tilde{P} \in \mathbb{R}^{n\times n}, \quad \tilde{R} = nI_m, \quad \tilde{R} \in \mathbb{R}^{m\times m},$$

$$\tilde{Q} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \\ \vdots & & & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix}, \quad \tilde{Q} \in \mathbb{R}^{n\times m}.$$

Thus, the generalized Hessian matrix consists of 4 blocks, two of which are diagonal matrices, and the two others have all elements equal to 1. This matrix will be called the *initial generalized* Hessian matrix. Then, at each iteration of the process of solving the unconstrained maximization problem (27), the generalized Hessian matrix is obtained by stepwise transformations of the original matrix $\overline{A}\overline{A}^{\top}$. If, on the diagonal of the matrix $D(z)$ in the ith position, the first zero element occurs, then, in the original generalized Hessian matrix, four elements are reduced by 1 as follows:

$$\tilde{p}_{kk} := \tilde{p}_{kk} - 1, \quad \tilde{r}_{dd} := \tilde{r}_{dd} - 1, \quad \tilde{q}_{k(m+d)} := 0, \quad \tilde{q}_{(m+d)k} := 0, \tag{29}$$

where

$$k = \lceil i/n \rceil, \quad d = \begin{cases} n, & i - n\lfloor i/n \rfloor = 0, \\ i - n\lfloor i/n \rfloor & \text{otherwise.} \end{cases}$$

Here, $f(x) = \lceil x \rceil$ is the upward rounding and $f(x) = \lfloor x \rfloor$ is the downward rounding of the argument to the nearest integer number. Further transformations are carried out step by step in accordance with the position of subsequent zero elements on the diagonal of the matrix $D(z)$. If, on the contrary, at some iteration, zero elements in the matrix $D(z)$ are replaced by units, then we perform the recalculation by similar formulas:

$$\tilde{p}_{kk} = \tilde{p}_{kk} + 1, \quad \tilde{r}_{dd} = \tilde{r}_{dd} + 1, \quad \tilde{q}_{k(m+d)} = 1, \quad \tilde{q}_{(m+d)k} = 1. \tag{30}$$

As a result, the computations of the generalized Hessian matrix can be simplified considerably by taking into account at each iteration only those elements of the matrix $D(z)$ that have changed. If, on passage to the $(k+1)$-th iteration, l elements of the matrix $D(z)$ have changed in comparison with the kth iteration, then, for calculating the matrix S_{pp}^{k+1}, we must perform $4l$ elementary transformations of the matrix S_{pp}^{k}.

Let us give an example that clearly demonstrates the transformations of the generalized Hessian matrix. Consider the case of $m = 2$ and $n = 3$. Then, the initial Hessian matrix takes the form:

$$S_{pp} = \begin{pmatrix} 3 & 0 & 1 & 1 & 1 \\ 0 & 3 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 & 2 \end{pmatrix}.$$

Let $D_2(z) = 0$ be the first (and only) zero element occurring on the diagonal of the matrix $D(z)$. Then, the matrix S_{pp} will be rewritten (in one step) as follows:

$$S_{pp} = \begin{pmatrix} 2 & 0 & 1 & 0 & 1 \\ 0 & 3 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 2 \end{pmatrix}.$$

Obviously, exactly four elements have changed, one in each block. Let also $D_3(z) = 0$. Now the matrix S_{pp} has the form

$$S_{pp} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Table 2. Computation results

$m \times n$	j, steps	Δ_1	Δ_2	t, s
10 × 20	8	4.0×10^{-13}	3.0×10^{-12}	0.07
	8	4.0×10^{-13}	3.0×10^{-12}	0.09
60 × 80	6	3.0×10^{-12}	4.5×10^{-13}	0.4
	6	3.0×10^{-12}	4.5×10^{-13}	0.64
100 × 200	6	1.0×10^{-11}	2.3×10^{-10}	3.4
	6	1.0×10^{-11}	2.3×10^{-10}	3.2
100 × 300	8	1.1×10^{-11}	1.5×10^{-10}	10.1
	8	1.1×10^{-11}	1.5×10^{-10}	7.3
250 × 400	10	2.0×10^{-11}	1.0×10^{-9}	102
	10	2.0×10^{-11}	1.0×10^{-9}	45
300 × 500	8	2.5×10^{-11}	1.6×10^{-9}	132
	8	2.5×10^{-11}	1.6×10^{-9}	70
200 × 600	7	2.9×10^{-11}	2.6×10^{-10}	138
	7	2.9×10^{-11}	2.6×10^{-10}	67
300 × 700	6	3.6×10^{-11}	7.6×10^{-10}	91
	6	3.6×10^{-11}	7.6×10^{-10}	47

Further on, if $D_5(z) = 0$, then

$$S_{pp} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

etc., in accordance with formulas (29) and (30).

In order to determine the efficiency of the approach considered, a series of numerical experiments was performed. In this case, we compared the generalized Newton's method, whose application to the linear programming problem was described in [1, 2], and its modification in accordance with the above formulas of iterative transformations of the generalized Hessian matrix. Transport problems generated with the given number of variables n and the given number m of constraints were solved. To calculate the coefficients of the objective function $\overline{c}$ and the right-hand sides $\overline{b}$ of the LP problem, we used the solutions of the primal and dual problems, $\overline{x}_*$ and $\overline{u}_*$, which were generated as follows: in the vector $\overline{x}_*$, the components were chosen randomly from the interval [0,10] and, for the vector $\overline{u}_*$, half of the components were set to zero and the rest were randomly chosen from the interval [0,10]. Then the vectors $\overline{b}$ and $\overline{c}$ were determined by the formulas

$$\overline{b} = \overline{A}\overline{x}^*, \quad \overline{c} = \overline{A}^\top \overline{u}^* + \xi,$$

where, if $\overline{x}_i^* > 0$, then $\xi_i = 0$, and, if $\overline{x}_i^* = 0$, then the component ξ_i was chosen randomly from the interval

$$0 \le \gamma_i \le \xi_i \le \theta_i.$$

After terminating the iterative process at the Kth step, the residuals were calculated, from which the correctness of the solution x^K found can be determined:

$$\Delta_1 = \left\| \bar{A}\hat{\bar{x}}^K - \bar{b} \right\|_\infty, \quad \Delta_2 = \left| \bar{c}^\top \hat{\bar{x}}^K - \bar{c}^\top \hat{\bar{x}}^* \right|.$$

Table 2 presents the results of the computational experiment. Each problem was solved using Newton's method. In the first case, the generalized Hessian matrix was calculated using formula (28) (upper line) and formulas (29) and (30) (bottom line). The first column indicates the number of variables and the number of constraints in the transport problem generated. The second column indicates the number of iterations in which the solution was found, and the last column gives the computation time. The third and fourth columns give the residuals. The problems were solved in MATLAB 2012b on a personal computer with a dual-core Intel (R) Core (TM) i5-2410M processor with a frequency of 2.30 GHz and 8 GB RAM.

The results show that the solutions were obtained with a high accuracy. In some cases, the modified algorithm reduces the computation time. This occurs in the case of a large number of changes in the diagonal elements of the matrix $D(z)$. In the last numerical experiment with the problem with the dimension 300×700, the step size was chosen using the Armijo rule.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research (project no. 17-07-00510), the Program for the Support of the Leading Scientific Schools (project no. NSh-8860.2016.1), and the Ministry of Education and Science of the Kazakhstan Republic (project no. 0115RK00554).

REFERENCES

1. A. I. Golikov and Yu. G. Evtushenko, "Solution method for large-scale linear programming problems," Dokl. Math. **70** (1), 615–619 (2004).
2. V. A. Garanzha, A. I. Golikov, Yu. G. Evtushenko, and M. Kh. Nguen, "Parallel implementation of Newton's method for solving large-scale linear programs," Comput. Math. Math. Phys. **49** (8), 1303–1317 (2009).
3. O. L. Mangasarian, "A Newton method for linear programming," J. Optim. Theory Appl. **121**, 1–18 (2004).
4. L. D. Popov, "Quadratic approximation of penalty functions for solving large-scale linear programs," Comput. Math. Math. Phys. **47** (2), 200–214 (2007).
5. L. D. Popov, "Experience in organizing hybrid parallel calculations in the Evtushenko–Golikov method for problems with block-angular structure," Autom. Remote Control **75** (4), 622–631 (2014).
6. I. I. Eremin, "On quadratic and fully quadratic problems of convex programming," Russ. Math. **42** (12), 20–26 (1998).
7. M. Frank and P. Wolfe, "An algorithm for quadratic programming," Naval Res. Logist. Quart. **3**, 95–110 (1956).
8. A. N. Tikhonov, "On ill-posed problems in linear algebra and a stable method for their solution," Dokl. Akad. Nauk SSSR **163** (4), 591–594 (1965).
9. V. N. Malozemov and G. Sh. Tamasyan, "Two fast algorithms for projecting a point onto the canonical simplex," Comput. Math. Math. Phys. **56** (5), 730–743 (2016).
10. G. Sh. Tamasyan, E. V. Prosolupov, and T. A. Angelov, "Comparative study of two fast algorithms for projecting a point to the standard simplex," J. Appl. Ind. Math. **10** (2), 288–301 (2016).

Translated by E. Chernokozhin