Proc. Estonian Acad. Sci. Phys. Math., 1998, **47**, 4, 251–259
https://doi.org/10.3176/phys.math.1998.4.01

ON COMPUTING A STABLE LEAST SQUARES SOLUTION TO THE LINEAR PROGRAMMING PROBLEM

Evald ÜBI

Tallinn Technical University, Ehitajate tee 5, 12618 Tallinn, Estonia; e-mail: evaldy@tv.ttu.ee

Received 26 February 1998, in revised form 16 April 1998

Abstract. The linear programming problem is transformed to the quadratic programming problem – to find the smallest distance. For the solution of this problem the least squares method is used. The problem is reduced to another one – to find a nonnegative solution of an overdetermined system of linear equations. The least squares method is recommended for solving problems with degenerate basis or with ill-conditioned matrix. A test problem with the Hilbert matrix is solved up to the 220th order, while most applications deal with 4th- to 10th-order matrices.

Key words: linear programming, method of least squares, ill-conditioned problems.

1. INTRODUCTION

Let A be an $m \times n$ matrix, b an m-dimensional, and c an n-dimensional vector. Let us consider the linear programming problem

$$\begin{aligned} z = (c, x) &\rightarrow \max, \\ Ax &= b, \\ x &\geq 0. \end{aligned} \tag{1}$$

Let $x*$ be a solution of the system (1), $z* = (c, x*)$, and $x(\epsilon)$ the least squares solution of the overdetermined system

$$\begin{aligned} Ax &= b, \\ \epsilon x &= c^T, \\ x &\geq 0, \end{aligned} \tag{2}$$

where T denotes the transpose, $\epsilon > 0$. In Section 4 it is proved that $x(\epsilon) \to x*$ if $\epsilon \to 0$. Denote $D_\epsilon = (A, \epsilon I)^T$ and $h = (b, c^T)^T$, where I is the unit matrix. Then (2) can be written as

$$\begin{aligned} D_\epsilon x &= h, \\ x &\geq 0. \end{aligned} \tag{3}$$

The problems (2) and (3) are equivalent to the problem

$$\Phi\epsilon(x) = \| h - D_\epsilon x \|^2 \to \min, \tag{4}$$

which after some transformations becomes

$$\Phi\epsilon(x)/\epsilon = (b - Ax, b - Ax)/\epsilon + (c, c)/\epsilon - 2(c, x) + \epsilon(x, x) \to \min. \tag{5}$$

Therefore, the solution of the problem (2) in least squares is equivalent to applying penalty functions and regularization methods to (1). The term $\epsilon(x, x)$ in (5) guarantees the stability of the method and enables us to solve unstable problems with great accuracy (see Section 3). Another way for using least squares is presented in [1]. Instead of the system (2) there are only $m + 1$ equations

$$\begin{aligned} Ax &= b, \\ \epsilon(c, x) &= \epsilon z_0, \\ x &\geq 0, \end{aligned} \tag{6}$$

for which the parameter $z_0 \geq z*$ is to be chosen, $\epsilon > 0$. However, the least squares solution of this system is not as precise as that of (2), because the later is found using a regularization method. In Section 2 a detailed description of the algorithm VL to solve (3) is given. It is based on the QR decomposition of D_ϵ which has two characteristic features (see [2]). First, the order of the active variables corresponding to the columns of the triangular matrix R is determined by the third step of the algorithm VL: the variable x_j is activized if the angle between the column d_j and h is smallest. Second, if in the solution process of the system with a triangular matrix R some variable proves to be nonpositive (see, e.g., Example 1, iteration 4), then the column corresponding to this particular variable is eliminated from the matrix R and the other columns are transformed again to the triangular form using Givens plane rotations. The order of the matrix R is equal to the number of the active variables: on the first step it is 1, on the second step 2, etc. These values are determined on each step from the triangular system with the matrix R. The algorithm VL is finite, because on each step the minimum of the function $\Phi_\epsilon(x)$ in some subspace is found. The number of these subspaces is finite (see [2]). A well-known stabilizing system $Ax = b, \epsilon x = 0$, which is composed for solving an exactly defined system of linear equations $Ax = b$, corresponds to the system (2).

2. DESCRIPTION OF THE ALGORITHM VL

Let us describe the algorithm for finding an approximate solution $x(\epsilon)$ to the problem (1). The vector $x(\epsilon)$ by fixed ϵ is found as a least squares solution of (2) or (3). The matrix D of the system (3) has $m1$ rows and n columns, h is an $m1$-dimensional vector, $m1 = m + n$. Furthermore, five more n-vectors – c, x, F, G, IJ and an m-vector u are needed.

Algorithm $VL(D, h, c, IJ, x, u, F, G, m1, n, \epsilon, \epsilon 1, M)$.

1. Take for the number of active variables $k = 0$ and $x = 0$.

2. Find n-vectors F and G with the coordinates

$$F(j) = (d(j), h), G(j) = (d(j), d(j)), \quad j = 1, ..., n.$$

3. Determine the following active variable $x(j_0)$ by computing

$$\max F^2(j)/G(j) = F^2(j_0)/G(j_0) = RE,$$

where maximum is found for all passive variables for which $G(j) > \epsilon 1$ and $F(j) > 0$.

4. If $RE < \epsilon 1$, then go to step 19.

5. Increase the number of active variables, $k = k + 1$, and store the index j_0 in the array IJ, $IJ(k) = j_0$.

6. Apply the Householder transformation to the columns $d(j)$ and the right side h with the vector $v = d(j_0)$, taking $m1 + 1 - k$ for the dimension of these vectors (see [3], Ch. 10).

7. Calculate new $F(j) = F(j) - d(k, j)h(k)$, $\quad G(j) = G(j) - d^2(k, j)$, $j = 1, ..., n$.

8. Calculate the values of the active variables x_j from the triangular system.

9. Take the number of the controlled variable $L = k + 1$. (During steps 9–13 positivity of active variables is checked.)

10. Let $L = L - 1$.

11. If $L = 0$, then go to step 12, or else go to step 13.

12. If $m \geq k$, then go to step 3, or else go to step 19.

13. If the inequality $x(j(L)) > 0$ holds, where $j(L)$ is the index of the active variable, then go to step 10.

14. Let $x(j(L)) = 0$ and delete index $j(L)$ from the array $IJ, IJ(i) = IJ(i + 1)$, $i = L, ..., k - 1$.

15. To delete the column $d(j(L))$ from the set of active columns, transfer the matrix which consists of active columns to the triangular form using Givens rotations. In order to eliminate $v(2)$ in the vector $v = (v(1), v(2))^T$, all two-dimensional vectors are multiplied by the Givens matrix G, where $G(1, 1) = G(2, 2) = c$, $\; G(1, 2) = s$, $\; G(2, 1) = -s$, $\; c = v(1)/\, sqrt(v(1)^2 + v(2)^2)$.

16. Find new $F(j) = F(j) + d(k, j)h(k)$, $\quad G(j) = G(j) + d(k, j)^2$, $j = 1, ..., n$.

17. Decrease the number of active variables, $k = k - 1$.
18. Go to step 8.
19. Check the inequalities $| h(i) | \leq \epsilon 1 + \epsilon((b,b)+(c,c))$ for $i = k+1, ..., m$. If any of these inequalities does not hold, then the problem has no feasible solutions; stop.
20. Find the value of the goal function $z = (c, x)$.
21. If $z > M$, then the goal function is unbounded; stop.
22. The problem is solved.

Example 1. Let us solve

$$\begin{array}{llll} x(1) & +3x(2) & +2x(3) & = z \to \max, \\ x(1) & +x(2) & +x(3) & = 3, \\ 2x(1) & & +3x(3) & = 6, \\ & & x & \geq 0 \end{array}$$

for $\epsilon = 0.01$ and $\epsilon 1 = 10^{-25}$.

Iteration		x(1)	x(2)	x(3)	h
1		1	1	1	3
		2	0	3	6
		0.01	0	0	1
		0	0.01	0	3
		0	0	0.01	2
	F	15.01*	3.03	21.02	
	G	5.0001*	1.0001	10.0001	
	x	0	0	0	
2		−2.2361	−0.4472	−3.1305	−6.7126
		0	−0.8944	0.4473	−0.0027
		0	−0.0045	−0.0128	0.9700
		0	0.0100	0	3
		0	0	0.0100	2
	F	0	0.0281*	0.0064	
	G	0	0.8001*	0.2003	
	x	3.0019	0	0	
3		−2.2361	−0.4472	−3.1305	−6.7126
		0	0.8945	−0.4471	0.0314
		0	0	−0.0150	0.9701
		0	0	0.0050	2.9998
		0	0	0.0100	2.0000
	F	0	0	0.0204*	
	G	0	0	0.0003	
	x	2.9949	0.0351	0	

4		−2.2361	−0.4472	−3.1305	−6.7126
		0	0.8945	−0.4471	0.0314
		0	0	0.0187	1.0930
		0	0	0	2.9816
		0	0	0	1.9635
	F	0	0	0	
	G	0	0	0	
	x	−84.6374	29.2411	0.0323	
5		1.0000	1.0001	1.000	3.0298
		2.0000	0	3.0000	5.9967
		−0.0125	0	0	1.0556
		0	0	0	2.9816
		0	0	0	1.9635
	F	−0.0132	0	0	
	G	0.0121	0	0	
	x	0	1.0310	1.9989	

First $x(1)$, then $x(2)$, and finally $x(3)$ is activized. After applying three times Householder transformations, we have on the next step $x(1) = -84.6 < 0$. Delete the first one from the set of active columns. Rotating the first and the second row, annihilate the element $d(2,2)$ according to step 15 of the algorithm, then annihilate $d(3,3)$ analogously. The approximate solution $x(\epsilon) = (0, 1.0310, 1.9989)^T$ differs little from $x* = (0,1,2)^T$.

ϵ	$x(1)$	$x(2)$	$x(3)$
0.1	0	1.29904	1.98811
0.001	0	1.03102	1.99893
0.0001	0	1.00031	1.99999
0.00001	0	1.00003	2.00000

It is well known that the solution of the problem (3) satisfies the normal equations $D^T D x = D^T h$,

$$\begin{aligned} (5+\epsilon^2)x(1) \quad &+x(2) &+7x(3) &= 15+\epsilon, \\ x(1) \quad &+(1+\epsilon^2)x(2) &+x(3) &= 3+3\epsilon, \\ 7x(1) \quad &+x(2) &+(10+\epsilon^2)x(3) &= 21+2\epsilon, \end{aligned}$$

the solution of which in least squares $x(\epsilon) = (0,\ (9+28\epsilon+3\epsilon^2+3\epsilon^3)/t,\ (18-\epsilon+21\epsilon^2+2\epsilon^3)/t)^T \to x* = (0,1,2)^T$, $\epsilon \to 0$, $t = 9+11\epsilon^2+\epsilon^4$. The cross product matrix $D^T D$ is often ill-conditioned and the result is quite inaccurate.

Remark 1. The main disadvantage of the algorithm VL is the large amount of memory capacity needed. One can use it to solve comparatively small unstable problems. For large-size problems the well-known least squares technique should be used.

Remark 2. If, for $x \geq 0$, $u* = \min(b - Ax, b - Ax) > 0$, the problem (1) has no feasible solutions. Let k be the number of active variables in the least squares solution $x(\epsilon)$ of (2). If the right sides $h(k+1), ..., h(m)$ are close to zero, the problem is solvable because $h(k+1)^2 + ... + h(m)^2 \rightarrow u*$ if $\epsilon \rightarrow 0$. In step 19 the condition for the approximate solution fits with the results of calculations. The solvability is determined more surely using the exact algorithm VRA, where the function $u(x) = (b - Ax, b - Ax)$ on the set $x \geq 0$ is minimized (see [2]).

Remark 3. Whether the goal function is bounded can be checked by the algorithm VRA after solving the problem $Ax = b, (c, x) = M, \quad x \geq 0$, where M is sufficiently large. According to computing experiments, there seem to be $m + 1$ active variables in the result (if rank $A = m$ and if the goal function is unbounded) analogously to the simplex method. Thereby some of these variables and the goal function depend essentially on ϵ, being proportional to $1/\epsilon$.

Remark 4. A lot of trouble is caused by the degenerate basis. Probably the algorithm VL is more suitable to solve such problems, because a disturbed problem (see Example 1, iteration 3) arises by adding the constraint $\epsilon x = c^T$. The choice of the weight ϵ is considered in Section 3.

Remark 5. Probably a number of problems can be solved more rapidly if, after finding a feasible basic solution, one continues with the simplex method. In this case one can continue by applying the algorithm $VRMSIM$ which uses Givens rotations and triangular basic matrix instead of Gaussian elimination (see [4]). More detailed analysis of this question is not the topic of the present article.

Remark 6. If the optimal solution of the problem (1) is not unique, it may turn out that $x(\epsilon)$ is not a basic solution but some vector with $m + 1$ positive components. This vector is not a vertex (see, e.g. Example 2).

Remark 7. During the actual solution process the equality $x(j) > 0$ held almost always.

3. NUMERICAL EXPERIMENTS

Determination of the weight ϵ is considered in [3]. Probably the constraint $x \geq 0$ in mathematical programming problems does not influence the choice of ϵ.

In the system (2) equations $Ax = b$ must exist first; otherwise one should rearrange rows in order to achieve stability. Also, the elements of the matrix A and vector b must be essentially greater than ϵ. However, for too small ϵ and due to limited accuracy of calculations, a solution $x(\epsilon)$ of the system (2) turns out to be the same as for $\epsilon = 0$. In other words, we get some feasible solution which does not depend on the vector c. Besides, the condition number of the system (2) can converge to infinity if $\epsilon \to 0$. As said above, the algorithm VL is essentially more precise than the algorithm VR used to find the least squares solution of (6). For both algorithms activizing of variables depends on the sign of the inner product $F(j)$ (see steps 2 and 3). In the algorithm VL, $F(j)$ is a linear function of the weight ϵ; in VR it is quadratic. In general, using the algorithm VR, the choice of ϵ is made in a smaller domain than in VL, and the accuracy of the solution is lower as well.

According to Theorem 1, in the next section the solution of the problem (2), $x(\epsilon)$, converges to the optimum solution which has the smallest norm. As for more than $m+1$ active variables it is difficult to guarantee the stability of the algorithm VL, then, based on step 12, the number of active variables k is always less than $m+2$ (see Example 2).

Example 2.

$$\begin{array}{rrrrrrrcl}
(1+t)x(1) & +x(2) & +x(3) & +x(4) & +x(5) & & & = & 4+t, \\
x(1) & & +x(3) & +x(4) & & +x(6) & & = & 3, \\
x(1) & & & +x(4) & & & +x(7) & = & 2, \\
x(1) & +x(2) & +x(3) & +x(4) & & & & = & z \to \max, \\
& & & & & & x \geq 0. & &
\end{array}$$

The maximum value of the goal function is $z* = 4+t$. In the case of $t = 0.00001$ and $\epsilon = 0.0000001$ the algorithm VL found $x(\epsilon) = (0,\ 1.0000101,\ 1.3333333,\ 1.6666667,\ 0,\ 0,\ 0.3333333)$.

Example 3. Let us consider a linear programming problem with the Hilbert matrix $d(i,j) = 1/(i+j)$, $h(i) = 1/(i+1) + 1/(i+2) + ... + 1/(i+m)$, $d(i, m+i) = 1$, $d(m+i, m+i) = \epsilon$, $h(m+i) = h(i) + 1/(i+1)$, $i, j = 1, ..., m$; the rest of the elements $d(i,j) = 0$. For $\epsilon = 0.00001$ the absolute error δ almost does not depend on the range of the system: $\delta = 0.012$ if $40 \leq m \leq 220$. For $220 < m$ the high-speed memory was not sufficient to store the matrix D. Well-known programs solve this problem with the Hilbert matrix only if m is in the interval [4,10]. In both examples $\epsilon 1 = 10^{-26}$. All computations were performed on an IBM-4381 using FORTRAN codes. For all variables double exactness was used.

4. PROOF OF THE CONVERGENCE

Suppose the $p \times n$ matrix E, the p-vector f, the $m \times n$ matrix A, and the m-vector b are given. We shall consider the quadratic programming problem of finding an n-vector $x*$ so that it minimizes the sum of squares

$$\begin{aligned}
v &= \| f - Ex \|^2 \to \min, \\
Ax &= b, \qquad\qquad (7) \\
x &\geq 0,
\end{aligned}$$

where $\|\cdot\|$ is the Euclidean norm. To solve this problem, let us consider the system

$$\begin{aligned} Ax &= b, \\ \epsilon Ex &= \epsilon f, \\ x &\geq 0, \end{aligned} \tag{8}$$

where its least squares solution is denoted by $x(\epsilon), \epsilon > 0$. In [1] it is proved that $x(\epsilon) \to x*$ if $\epsilon \to 0$. Denote by X^* the set of optimal solutions of (1). Assume that X^* is not empty and prove the obvious statement: The linear programming problem (1) is equivalent to finding a feasible solution, the nearest to a sufficiently distant point tc^T, where t is a large positive parameter and T denotes the transpose. Let us denote the optimal solution of the problem

$$\begin{aligned} v &= \| x - tc^T \|^2 \to \min, \\ Ax &= b, \\ x &\geq 0 \end{aligned} \tag{9}$$

by $x_t, \quad t > 0$.

Theorem 1. *There exists a number t_0 such that for each $t \geq t_0$ the equality $x_t = x*$ holds, where $x*$ is the normal solution of* (1), *i.e. it is an element of X^* having the smallest norm.*

Transfer the goal function of (9),

$$-v/t = -(x, x)/t + 2(c, x) - t(c, c) \to \max .$$

According to [5] (Theorem 1, Par. 3, Ch. 10), there exists a number t_0 such that for each $t \geq t_0$ the equality $x_t = x*$ holds.

If the problem (1) has no solutions, the algorithm VL finds approximately a nonnegative solution of the system $Ax = b$ in least squares.

Theorem 2. *The solution in least squares of the system* (2) *converges to the normal solution $x*$ of the problem* (1),

$$\lim x(\epsilon) = x*, \ \epsilon \to 0. \tag{10}$$

Form an overdetermined system (8) (corresponding to the problem (9))

$$\begin{aligned} Ax &= b, \\ \epsilon x &= \epsilon tc^T, \\ x &\geq 0, \end{aligned}$$

where $t \geq t_0$ (see Theorem 1). The last system is equivalent to the system (2) if $t = 1/\epsilon$. This is valid for arbitrary $\epsilon \leq \epsilon_0 = 1/t_0$. If the problem (1) has a unique optimal solution, then $x(\epsilon) \to x*, \ \epsilon \to 0$.

REFERENCES

1. Übi, E. An approximate solution to linear and quadratic programming problems by the method of least squares. *Proc. Estonian Acad. Sci. Phys. Math.*, 1998, **47**, 1, 19–28.
2. Übi, E. Finding non-negative solution of overdetermined or underdetermined system of linear equations by method of least squares. *Trans. Tallinn Tech. Univ.*, 1994, **738**, 61–68.
3. Lawson, C. and Hanson, R. *Solving Least Squares Problems*. Prentice-Hall, New Jersey, 1974.
4. Übi, E. Conversion of the simplex tableau by the Givens rotations. *Proc. Estonian Acad. Sci. Phys. Math.*, 1991, **40**, 2, 75–79.
5. Poljak, B. *Vvedenie v optimizatsiyu*. Nauka, Moscow, 1983.

LINEAARSE PLANEERIMISE ÜLESANDE STABIILSE LAHENDI LEIDMINE VÄHIMRUUTUDE MEETODIGA

Evald ÜBI

Käsitletav ülesanne taandub ülemääratud lineaarse võrrandisüsteemi mittenegatiivse lahendi leidmisele. Selleks on kasutatud vähimruutude meetodit, mis on stabiilsem kui Gaussi elimineerimisel põhinev simpleksmeetod.