Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Method of Solution for Quadratic Programs

C. E. Lemke,

A METHOD OF SOLUTION FOR QUADRATIC PROGRAMS*

C. E. LEMKE

Rensselaer Polytechnic Institute, Troy, New York

This paper describes a method of minimizing a strictly convex quadratic functional of several variables constrained by a system of linear inequalities. The method takes advantage of strict convexity by first computing the absolute minimum of the functional. In the event that the values of the variables yielding the absolute minimum do not satisfy the constraints, an equivalent and simplified quadratic problem in the 'Lagrange multipliers' is derived. An efficient algorithm is devised for the transformed problem, which leads to the solution in a finite number of applications. A numerical example illustrates the method.

1. Introduction

We are concerned with the general quadratic programming problem posed in the following form: Minimize

$$-a^T x + \tfrac{1}{2}x^T Q_0 x, \tag{1}$$

over the set of all x satisfying:

$$A^T x \leq d_0\,. \tag{2}$$

Here matrix A has order $m \times n$; Q_0 is $m \times m$; a and x are m $\times$ 1; and d_0 is $n \times 1$. Superscript 'T' denotes matric transposition. It is assumed that Q_0 is positive definite and symmetric.

Ignoring constraints (2) momentarily, the minimum of (1) is taken on at the unique point

$$x_0 = Q_0^{-1} a, \tag{3}$$

where the gradient $Q_0 x - a$ vanishes. If, further, x_0 satisfies the constraints (2), it must then solve the quadratic problem. If not, the optimum is taken on for some point x on the boundary of the convex polyhedron described by the constraints (2).

In a recent article [7], Theil and van de Panne appear to have been the first to utilize the non-singularity of Q_0, and, starting from knowledge of x_0, to systematically search out the optimal boundary point. As the authors noted, such a technique does not require a feasible point to initiate the calculations. Their technique is compared briefly at the end of this paper with the one described below. Actually our method appears to resemble more that proposed by Beale in 1955 [1].

In Section 2, we derive an equivalent problem, which is immediately feasible and permits a simple algorithm. Following Section 3, where the algorithm is

* Received August 1961.

described and necessary proofs given, an example, also used by Theil and van de Panne, is furnished.

2. Initial Transformation

The Kuhn-Tucker geometric conditions for an optimum state that a feasible point x is a solution if and only if the gradient $(Q_0x - a)$ at x is a non-positive combination of the outward-directed normals of those support hyperplanes, if any, containing x; that is, if and only if there is some y such that:

$$(4) \quad (Q_0x - a) + Ay = 0; \qquad y \geqq 0; \qquad \text{and} \quad y^T(d_0 - A^Tx) = 0;$$

the last condition ensuring that only those support planes containing x are considered. $y = 0$ would correspond to the point x_0, which would be the unique solution if it satisfied the constraints.

In the first part of (4) we may solve for x in terms of y uniquely:

$$(5) \qquad x = Q_0^{-1}(a - Ay),$$

from which the final solution may be computed, using an optimal set y of 'Lagrange multipliers'.

Using (5) we may eliminate x from (2). Then defining:

$$(6) \qquad Q = A^TQ_0^{-1}A; \qquad d = d_0 - A^Tx_0; \qquad \text{and} \quad z = d + Qy,$$

we may pose the following problem which, because of (5), is fully equivalent to the original:

Find a pair of vectors y and z which satisfy:

$$(7) \qquad -Qy + z = d; \qquad y, z \geqslant 0; \qquad \text{and} \quad y^Tz = 0,$$

where Q is symmetric and non-negative definite.

Let us now note that, by the Kuhn-Tucker conditions, this problem is entirely equivalent to the following quadratic problem in the variable point y:

Minimize

$$(8) \qquad f(y) = d^Ty + \tfrac{1}{2}y^TQy, \quad \text{subject to } y \geqslant 0,$$

where Q and d represent the given data, and Q is a symmetric and non-negative definite matrix.

It is this 'derived' problem, which has some interest in its own right, which shall be solved. Having an optimal y, the optimal x for the original problem is obtained as in (5).

Note that a 'feasible' point, i.e., one satisfying the constraints $y \geqslant 0$; namely $y = 0$, yielding $f(0) = 0$, is immediately available, and is indeed the only extreme point of the constraint set (the non-negative orthant), which is just a simple cone. $y = 0$ would yield the solution if and only if $d \geqslant 0$, as (7) shows.

Finally, with reference to the method to be described, note that no assumptions as to strict convexity of $f(y)$, or any assumptions as to 'degeneracy' usually required for linear programming methods are made. We require merely that a finite solution exists.

3. The Algorithm

Preliminaries

With reference to the conditions (7) for an optimum, y and z form an optimal pair if and only if (i) y is a point in the non-negative orthant; (ii) z, the gradient of f at y, points into the non-negative orthant; and (iii) y is perpendicular to its corresponding gradient z.

Throughout the calculations condition (i), starting with $y = 0$, is retained, and as the iterations proceed the functional f will be non-increasing. After a finite number of iterations, the calculations will end with conditions (ii) and (iii) satisfied.

Associated with an iteration is a point y and a set of n independent directions. As in the extreme point methods of linear programming, these directions appear as the rows of the inverse of an $n \times n$ 'current basis matrix'. In going from one iteration to the next, the current basis matrix is altered by replacing a single one of its columns by some other column, and the required 'current inverse' is obtained by algorithm from the inverse associated with the previous iteration. The original data Q and d are retained throughout, and all calculation is based on the computed inverse.

We adopt the following notation. If B is the current basis matrix we write:

$$B = (b_1, b_2, \cdots, b_n), \tag{9}$$

so that b_i denotes the i^{th} column of B. Further we write:

$$(B^{-1})^T = (b^1, b^2, \cdots, b^n), \tag{10}$$

so that b^i denotes the i^{th} row of B^{-1}, written in column form. The statement that $BB^{-1} = I$, the identity matrix of order n, is then equivalent to the n^2 scalar conditions:

$$b_i^T b^i = 1; \qquad b_i^T b^j = 0; \qquad \text{for} \quad i \neq j. \tag{11}$$

We denote the i^{th} column of I by e_i, which is thus a column with 1 as its i^{th} component and 0 as its other components. The constraints $y \geqslant 0$ may be expressed in scalar form as $e_i^T y \geqslant 0$; $i = 1, 2, \cdots, n$.

Now as to the composition of the current basis matrix B, during an iteration some components of the current feasible point y will be 0; that is, for some values of i, y will lie on the 'bounding hyperplane' $e_i^T y = 0$, with positive normal e_i. For some of these i (initially, all of them) the corresponding vector e_i will appear as a column of B. If for any i, we have e_i as a column of B and also $e_i^T y = 0$ we shall (following Beale [1]; see also [6]) call such a column of B a 'restricted' column. On the first iteration, all columns of B are restricted columns and in fact $B = I$. The other columns of B are 'free' columns, and are generated by the algorithm as described below.

Thus, for each iteration there will be a subset R of the set of integers from 1 to n which will specify which columns of B are restricted columns.

The current basis B will change by one column in going from one iteration to the next. We will denote by b_r the column to be replaced, and by $\underset{\sim}{b}_r$ the replacing column.

Finally, the criterion for continuing the iterations is based on the values of the current w expressed by:

$$w = B^{-1}z = B^{-1}(d + Qy). \tag{12}$$

One Iteration

At the start of the iteration one has the following computed data:

a. B^{-1} (whose i^{th} row is b^{i^T})
b. $y \geqslant 0$ (the current feasible point)
c. z (the gradient of f at y)
d. R (specifying which columns of B are restricted).

The above set of data is obtained by algorithm from the previous iteration's data. In the following, underscored items will refer to the new iteration.

To initiate the calculations, $B = B^{-1} = I$, so that $b^i = b_i = e_i$; $y = 0$; $w = z = d$; and R is the whole set. An iteration consists of selecting a column of B to be replaced; selecting a column subsequently to replace it; and modifying the data.

Selecting b_r Compute: $w = B^{-1}z$.

If the pair of current solutions y and z are not optimal there are two cases:

Case I. Some component of w corresponding to a free column of B is not zero.

Case II. All components of w corresponding to free columns of B are zero, but $w \geqslant 0$ does not hold.

If neither of these cases hold we are finished, as will be shown. If either case holds there will be some value of i, which we label r, which singles out the r^{th} component ω_r of w. In Case I, ω_r is not 0, while b_r is a free column. In Case II, ω_r is negative, while b_r is a restricted column. In either case, b_r is selected as the vector to be replaced. In the event that more than one value of i qualifies for $i = r$, we may select any value. However, for the sake of definiteness we shall select that value of r for which ω_r has the largest absolute value.

Selecting $\underset{\sim}{b}_r$. We seek a new feasible point of the form:

$$\underset{\sim}{y} = y - \theta b^r, \tag{13}$$

where θ is so selected as to minimize f on that part of the line $y - \theta b^r$ which remains in the constraint set $y \geqslant 0$. The selection of θ will determine $\underset{\sim}{b}_r$. Thus we determine θ as follows: Compute:

$$q = Qb^r, \tag{14}$$

Now $q = 0$ or not. First suppose $q \neq 0$, and compute:

$$\theta_0 = \omega_r / q^T b^r. \tag{15}$$

The functional is minimized for this value of θ. In fact one easily verifies that:

$$\begin{aligned} f(y - \theta b^r) &= f(y) - \tfrac{1}{2} q^T b^r [\theta_0^2 - (\theta_0 - \theta)^2] \\ &\qquad (= f(y) - \theta\omega_r \quad \text{when} \quad q = 0). \end{aligned} \tag{16}$$

To retain feasibility, we then compute:

$$t_0 = \text{Min.}\ \frac{e_i^T y}{\theta_0\, e_i\, T_b^r}, \tag{17}$$

where the minimum is taken over those i for which the denominator is positive. If there is no such i we take t_0 as infinite.

We take θ equal to θ_0 if $t_0 > 1$, and θ equal to $t_0\theta_0$ if $t_0 \leq 1$. In the latter case it is possible that θ equals 0.

When $\theta = t_0\theta_0$ there is some value of i, which we label k, such that:

$$\theta = \frac{e_k^T y}{e_k^T b^r}. \tag{18}$$

If more than one value of k is possible, we select any one arbitrarily.

If $q = 0$, then:

$$f(y - \theta b^r) = f(y) - \theta(d^T b^r) = f(y) - \theta\omega_r,$$

So that any non-zero θ having the sign of ω_r will decrease f. Then, from our assumption of finite optimum, for some θ having the sign of ω_r, $y - \theta b^r$ will strike the boundary of the set $y \geqslant 0$. Thus, when $q = 0$ we take θ_0 in (17) as signum ω_r, and $\theta = t_0\theta_0$.

The value of θ then defines four cases and the replacing vector as follows:

Case Ia or Case IIa: $\theta = \theta_0$. Then $\underset{\sim}{b}_r = Qb^r = q$

Case Ib or Case IIb: $\theta = e_k^T b^r$. Then $\underset{\sim}{b}_r = e_k$.

Modifying the data:

a. $\underset{\sim}{B}^{-1}$, whose rows are given by the formulas:

$$\underset{\sim}{b}^r = (1/\underset{\sim}{b}_r^T b^r)b^r; \qquad b^i = b^i - (\underset{\sim}{b}_r^T b^i)b^r \quad \text{for} \quad i \neq r. \tag{19}$$

b. $\underset{\sim}{y} = y - \theta b^r$.

c. $\underset{\sim}{z} = z - \theta q$.

d. $\underset{\sim}{R}$ is obtained from R by (i) no change for Cases Ia or IIb; (ii) including r for Case Ib; and (iii) deleting r for Case IIa.

This completes the description of an iteration.

Proof of Convergence

Recall that we are considering the system (7), and that we may refer to the minimization of $f(y)$ as in (8). We note again that the only assumption made is that a solution exists (or equivalently, that the minimum of $f(y)$ for $y \geqslant 0$ is finite). The lines of proof follow quite analogously those given in [6], but for the sake of the differences are repeated here.

Now the minimum is taken on either in the interior of the positive orthant or on its boundary. It is taken on in the interior if and only if there is a point $y > 0$ for which the gradient $d + Qy$ at y vanishes. If in the course of the iterations some such point is arrived at we are finished. That is, if ever the set R associated with an iteration is the empty set, and Case II occurs (i.e., $z = 0$) then we are finished. If the set R is not empty, it specifies the boundary of the positive orthant we are working with. Now (12) may be written as:

$$z = \sum_{i=1}^{n} \omega_i b_i , \tag{20}$$

expressing the gradient of f at y in terms of the current basis. Suppose that $\omega_i = 0$ for each i corresponding to a free column. Then (20) expresses z only in terms of columns of I, and further, $y^T w = 0$. Hence, by the Kuhn-Tucker conditions, (or in this case by the classical Lagrange multiplier conditions), the current y solves the problem:

$$\text{Min. } f(y) \text{ subject to } e_i^T y = 0, \quad \text{for all } i \text{ in } R; \tag{21}$$

that is, y yields the minimum over the face defined by R. If further we have $w \geqslant 0$ for the current w then, again, the Kuhn-Tucker conditions show that y solves the problem:

$$\text{Min. } f(y) \text{ subject to } e_i^T y \geqslant 0, \quad \text{for all } i \text{ in } R, \tag{22}$$

and hence, a fortiori, solves the problem (8). This will be the case when neither Case I nor Case II occurs. If $w \geqslant 0$ does not hold, then we have Case II. Thus, whenever Case II occurs the current y minimizes f on that face of the boundary specified by R. Thus, in particular, whenever Case II again occurs, and in the interim the functional f has been decreased, f will have been minimized on a different face, represented by a different R, (or, as we shall say, a better R). Now in going from one iteration to the next, f is not increased, so that one only goes to better R's. Since the number of sets R is finite, only a finite number of sequences of events: Case II—decrease in f—Case II is possible.

Convergence of the process will therefore follow when the following facts are demonstrated:

Lemma 1: When Case I occurs, Case II will occur in a finite number of iterations, unless the optimum is reached.

Lemma 2: When Case II occurs, there will be a decrease in f followed by a recurrence of Case II in a finite numbeı of iterations, unless the optimum is reached.

Proof of Lemma 1

When Case Ia occurs, the number s of free columns remains fixed. When Case Ib occurs, the number s of free columns is decreased by one. Therefore, if we show that when Case Ia occurs with s free columns it can continue to occur for at most s consecutive iterations, the recurrence of Case I can only continue to the situation where $s = 0$. But this corresponds to the initial iteration $y = 0$, with $f(0) = 0$. But, unless $d \geqslant 0$, in which case we are finished at the initial iteration, this situation is impossible. Thus we will have shown that consecutive occurrence of Case I must lead to Case II or the optimum.

Now when Case Ia occurs, with b_r as the vector being replaced, and with $\underline{b}_r = Qb^r$ as the replacing vector, we show that the r^{th} component of the new w is zero, and that when Case Ia continues to occur, those components of w which have become 0 in this way remain 0.

For this, consider a basis B with the property that for some i we have $b_i = KQb^i$ for some scalar K. Since for $j \neq i$ we have $0 = b_i^T b^j = b^{iT} Qb^j$, if for some $r \neq i$ we have a replacing vector $\underline{b}_r = Qb^r$, as in Case Ia, the new i^{th} row of the inverse remains unchanged: $\underline{b}^i = b^i - (\underline{b}_r^T b^i)\underline{b}^r = b^i$.

Now consider z as in (20). The new z is given by:

$$\underline{z} = z - \theta_0 Qb^r = \sum_{i=1}^{n} [\omega_i - \theta_0(b^{iT}Qb^r)]b_i. \tag{22}$$

The new coefficient of b_i in this expression is the i^{th} component of $\underline{w}$. The coefficient of b_r is 0 by definition of θ_0, and for each i such that we had (i) $\omega_i = 0$ and (ii) $b_i = KQb^i$ for some constant K we retain (i) $\underline{\omega}_i = 0$ and (ii) $\underline{b}^i = b^i$.

Thus, when Case Ia continues to occur on consecutive iterations, some additional component of w corresponding to a free column will become and remain 0. Hence Case Ia cannot continue to occur for more than s iterations. This proves the lemma.

Proof of Lemma 2

Suppose Case II occurs. Consider Case IIa. The functional is definitely decreased, since by (16) we then have:

$$f(\underline{y}) = f(y) - \tfrac{1}{2}(b^{rT}Qb^r)\theta_0^2. \tag{23}$$

Note that since Q is non-negative definite and symmetric, $a^T Qa = 0$ if and only if $Qa = 0$. Now on the next iteration either the optimum is obtained or not. If Case I occurs, Lemma 1 shows that either Case II occurs in a finite number of iterations or else the optimum is obtained. In either case, the sequence Case II—decrease in f—Case II occurs in a finite number of iterations, unless optimality is obtained.

Consider Case IIb. It is possible that in this case there is no decrease in f in proceeding to the next iteration. We show that Case I always occurs on the iteration following this case. With z expressed as in (20) we are supposing that all coefficients of b_i for i corresponding to a free column are 0. Since z does not change when θ is 0, and $\underline{b}_r = e_k$, we have:

$$z = \sum_{i=1}^{n} [\omega_i - (\omega_r/e_k^T b^r)e_k^T b^i]b_i + (\omega_r/e_k^T b^r)e_k, \tag{24}$$

expressing z in terms of the new basis, and specifying the new components of $\underline{w}$. Since e_k cannot be expressed in terms only of the restricted columns (which are all columns of I), some component $e_k^T b^i$ for which the i^{th} column of B is free is not 0, and the corresponding component of b_i in (24) is not 0. Hence Case I occurs on the next iteration.

Now if Case Ia occurs there is, by (23), a definite decrease in f, whereas if Case Ib occurs it is possible that still $\theta = 0$. But when Case Ib occurs, the number s of free columns is decreased by one. Now the only way to retain no decrease in f is to have a sequence of iterations with Case Ib and Case IIb only occurring, and each time with $\theta = 0$. But since Case Ib always follows Case IIb in this situation, and since s is decreased each time Case Ib occurs, this sequence must terminate with either the optimum, a definite decrease (Case Ia or IIa), or $s = 0$, which, as noted in the proof of Lemma 1, is impossible.

Hence, in either case, when Case IIb occurs, either the optimum or a decrease in f will follow which will, by Lemma 1, be followed either by the optimum or by Case II in a finite number of iterations. This proves the lemma.

4. An Example

The following example is used by Theil and van de Panne [7]. We shall take advantage of their computation of Q. We first consider the problem (8), with Q and d as the given data, and then return to the original problem.

We are solving the problem (8), where:

$$Q = \begin{bmatrix} 1.043 & -0.128 & -0.516 & 0.187 & -0.586 & -0.051 & -0.426 \\ -0.128 & 0.250 & 0.064 & -0.108 & -0.078 & -0.007 & -0.457 \\ -0.516 & 0.064 & 0.379 & -0.127 & 0.200 & -1.211 & 0.377 \\ 0.187 & -0.108 & -0.127 & 0.256 & -0.208 & 0.333 & -0.846 \\ -0.586 & -0.078 & 0.200 & -0.208 & 0.673 & 0.936 & 1.352 \\ -0.051 & -0.007 & -0.211 & 0.333 & 0.936 & 12.363 & -1.636 \\ -0.426 & -0.457 & 0.377 & -0.846 & 1.352 & -1.636 & 6.056 \end{bmatrix}$$

and:

$$d^T = (4.560 \quad 0.475 \quad -1.229 \quad 1.981 \quad -4.119 \quad -8.508 \quad -8.802)$$

Thus, initially, we have:

a. $B = B^{-1} = I$

b. $y = 0$

c. $z = d$

d. $R = \{1, 2, 3, 4, 5, 6, 7\}$.

Iteration 1

Selecting b_r :

(i) $$w = B^{-1}z = z = d.$$

Since no columns are free, *Case II* applies. Since $\omega_7 = -8.802$ is the most negative component of w, $b_r = e_7$.

Selecting b_r :

(ii) $$q = Qb^r = Qe_7 :$$

$$q^T = (-0.426 \quad -0.457 \quad 0.377 \quad -0.846 \quad 1.352 \quad -1.636 \quad 6.056)$$

(iii) $$q^T b^r = q^T e_7 = 6.056,$$

(iv) $$\theta_0 = \omega_7 / q^T b^r = -1.45343.$$

Since $\theta_0 < 0$, and $y - \theta b^r = -\theta e_7 \geq 0$ for any $\theta < 0$, we take t_0 infinite. Hence *Case IIa* applies and $\theta = \theta_0$ and $\underline{b}_r = q$.

Modifying the data:

a. $$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0.07035 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0.07546 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0.06225 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0.13970 \\ 0 & 0 & 0 & 0 & 1 & 0 & -0.22326 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.27015 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.16513 \end{bmatrix}$$

obtained using formulas (19).

b. $y = 0 - \theta_0 e_7$

$$y^T = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.45345).$$

c. $z = d - \theta_0 q$

$$z^T = (3.94084 \quad -0.18922 \quad -0.68106 \quad 0.75140 \quad -2.15396 \quad -10.88581 \quad 0)$$

d. $R = \{1, 2, 3, 4, 5, 6\}$.

Iteration 2

Selecting b_r :

(i) $w = B^{-1}z = z$ (since the last component of z equals 0).

Since column 7 is the only free column, and the 7th component of w is 0, *Case II* again applies. Since $\omega_6 = -10.88581$ is the most negative component of w, $b_r = e_6$.

Selecting $\underline{b}_r$:

(ii) $$q = Qb^6 \neq 0:$$

$$q^T = (-0.16608, \quad -0.13046, \quad -1.10915, \quad 0.10445, \quad 1.30123, \quad 11.92103, \quad 0).$$

(iii) $$q^T b^6 = q^T e_6 = 11.92103,$$

(iv) $$\theta_0 = \omega_6/q_6 = -0.91316.$$

Since $\theta_0 < 0$, $y - \theta b^r \geq 0$ again for any $\theta < 0$, and we take t_0 infinite. Hence *Case IIa* applies and $\theta = \theta_0$ with $\underline{b}_r = q$.

Modifying the data:

a. $$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0.01393 & 0.07411 \\ 0 & 1 & 0 & 0 & 0 & 0.01094 & 0.07842 \\ 0 & 0 & 1 & 0 & 0 & 0.09305 & -0.03712 \\ 0 & 0 & 0 & 1 & 0 & -0.00876 & 0.13733 \\ 0 & 0 & 0 & 0 & 1 & -0.10916 & -0.25275 \\ 0 & 0 & 0 & 0 & 0 & 0.08389 & 0.02266 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.16513 \end{bmatrix}$$

b. $y^T = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.91316 \quad 1.70012)$

c. $z^T = (3.78918 \quad -0.30835 \quad -1.69389 \quad 0.84678 \quad -0.96573 \quad 0 \quad 0)$
d. $R = \{1, 2, 3, 4, 5\}$.
Iteration 3

Selecting b_r :

(i) $w = B^{-1}z = z$ (again since the last two components of z are 0. This is due to the recurrance of Case IIa).

Since columns 6 and 7 oı B are free columns, and the corresponding components of w are 0; and since w is not yet non-negative, *Case II* again applies. Since $\omega_3 = -1.69389$ is the most negative component of w, $b_r = e_3$.

Selecting $\underline{b}_r$:

(ii) $q^T = (-0.50494 \quad 0.08031 \quad 0.25223 \quad -0.06461 \quad 0.23690 \quad 0 \quad 0)$.

(iii) $$q^T b^3 = q^T e_3 = 0.25223.$$

(iv) $$\theta_0 = -6.71566. = \omega_3/q^T e_3$$

Using formula (17) we have:

(v) $$t_0 = \frac{e_7{}^T y}{\theta_0 \, e_7{}^T b^3} = 6.82 > 1,$$

so that *Case IIa* applies; $\theta = \theta_0$; and $\underline{b}_3 = q$.

Modifying the data:

a. $$B^{-1} = \begin{bmatrix} 1 & 0 & 2.00191 & 0 & 0 & 0.20021 & -0.00020 \\ 0 & 1 & -0.31840 & 0 & 0 & -0.01869 & 0.09024 \\ 0 & 0 & 3.96464 & 0 & 0 & 0.36891 & -0.14717 \\ 0 & 0 & 0.25616 & 1 & 0 & 0.01508 & 0.12782 \\ 0 & 0 & -0.93922 & 0 & 1 & -0.19655 & -0.21789 \\ 0 & 0 & 0 & 0 & 0 & 0.08389 & 0.02266 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.16513 \end{bmatrix}$$

b. $y^T = (0 \quad 0 \quad 6.71566 \quad 0 \quad 0 \quad 1.53805 \quad 1.45083)$.
c. $z^T = (0.39817 \quad 0.23098 \quad 0 \quad 0.41288 \quad 0.62521 \quad 0 \quad 0)$.
d. $R = \{1, 2, 4, 5\}$.
Iteration 4

Selecting b_r

(i) $w = B^{-1}z = z \geq 0$; hence the solution has been found in three iterations. The fact that in all of the iterations only Case IIa occurred before the optimum was reached is due to the simplicity of the problem.

The values of the functional $f(y)$ may be obtained by algorithm using formula (16), or the final value only may be computed directly when the optimum has been reached. In the latter case, noting (7) we may write:

(24) $$f(y) = \tfrac{1}{2}y^T d + \tfrac{1}{2}y^T(d + Qy) = \tfrac{1}{2}y^T d + \tfrac{1}{2}y^T z = \tfrac{1}{2}y^T d.$$

The values of the functional for each iteration, starting with $f(0) = 0$ are 0, -6.397, -11.367, and -17.055.

We next return to the original problem which gave rise to the above example. This is an example of the problem (1) and (2), where

$$Q_0 = \tfrac{1}{2}\begin{bmatrix} 6 & 1 & 8 & 0 \\ 1 & 10 & 1 & 4 \\ 8 & 1 & 17 & 3 \\ 0 & 4 & 3 & 11 \end{bmatrix} \quad a = \begin{pmatrix} 9 \\ 8 \\ 11 \\ 10 \end{pmatrix}, \quad \text{and}$$

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 5 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & -1 & 0 & 1 & 10 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 5 \end{bmatrix}$$

$$d_0^T = (0 \quad 0 \quad 0 \quad 0 \quad \tfrac{5}{3} \quad 2 \quad 3).$$

One initially computes

$$x_0 = Q_0^{-1}a = \begin{pmatrix} 4.560 \\ 0.475 \\ -1.229 \\ 1.981 \end{pmatrix},$$

and subsequently $d = d_0 - A^T x_0$, which is the d used above, and which shows that x_0, the absolute minimum of the functional does not satisfy the constraints. One then computes $Q = A^T Q_0^{-1} A$, and solves the problem as above. Q_0^{-1} in this case appears as the 4×4 matrix in the upper left of Q.

Finally, it remains to calculate the solution x via formula (5):

$$x = x_0 - (Q_0^{-1}A)y = \begin{pmatrix} 0.400 \\ 0.233 \\ 2 \\ 0.414 \end{pmatrix},$$

with functional value given by

$$-a^t x + \tfrac{1}{2}x^t Q_0 x = -\tfrac{1}{2}a^t x_0 + y^t d = -17.037.$$

5. Discussion

It appears to be well worthwhile to take advantage of the non-singularity of the strictly quadratic part of the functional, when such is the case, to examine first the point x_0 where the absolute minimum of the functional is taken on. This allows the transformation to the problem (8), which could hardly be simpler in form, and the extremely simple algorithm described above.

Note that (i) there is no need to seek out a 'first feasible solution' for the constraints, as required for the approaches suggested by Beale [1] and Wolfe [8], (ii) there is no need to consider any 'degeneracy' cases.

The formulation (8), involving Lagrange multipliers as it does, is a form of dual problem to the original, as described by Dorn [4]. It is in fact Dorn's 'Type

II' dual, after eliminating the set of variables not constrained to be non-negative.

As noted, the algorithm is similar to the efficient one proposed by Beale in 1955. Perhaps the chief difference is that the initial data, Q is retained in its original form throughout the calculations, which are based on a 'current basis matrix', as in the modified simplex method [3] for linear programming.

Perhaps the method here suggested is more efficient than that proposed by Theil and van de Panne, although a general statement to that effect is out of the question. Two possible objections to the Theil-van de Panne approach as compared with ours are that (1) they do not make efficient use of computed data, but compute afresh inverses of submatrices of Q_0, and that (2) for large-order problems, the amount of data they compute and refer to seems to grow somewhat combinatorially.

An acknowledgement is in order. Theil and van de Panne base their technique on some ingeniously derived rules, which were subsequently shown by J. C. G. Boot [2] to be derivable from the Kuhn-Tucker conditions. It was Boot's observation which provided the stimulus for the present method.

References

1. Beale, E. M. L., "On Minimizing a Convex Function Subject to Linear Inequalities," *Journal of the Royal Statistical Society*, (Series B) 17: 173–184 (1955).
2. Boot, J. C. G., "Notes on Quadratic Programming: The Kuhn-Tucker Conditions, Degeneracy, and Equality Constraints," *Management Science*, Vol. 8, No. 1, pp. 85–98, (October 1961).
3. Charnes, A. and C. E. Lemke, "A Modified Simplex Method for Control of Round-off Error in Linear Programming," Proc. Assn. for Computing Machinery, Pittsburgh, May 1952.
4. Dorn, W. D., "Duality in Quadratic Programming," *QAM*, Vol. 18, No. 2, (July 1960).
5. Kuhn, H. W. and A. W. Tucker, "Non-Linear Programming," Proc. 2nd Berkeley Symp. on Math Stat. and Probability, University of California Press (1950).
6. Lemke, C. E., "The Constrained Gradient Method of Linear Programming," *SIAM Journal*, Vol. 9, No. 1, (March 1961).
7. Theil, H. and C. van de Panne, "Quadratic Programming as an Extension of Classical Quadratic Maximization," *Management Science*, Vol. 7, No. 1, (October 1960).
8. Wolfe, P., "The Simplex Method for Quadratic Programming," *Econometrica*, Vol. 27, pp. 382–398 (1959).