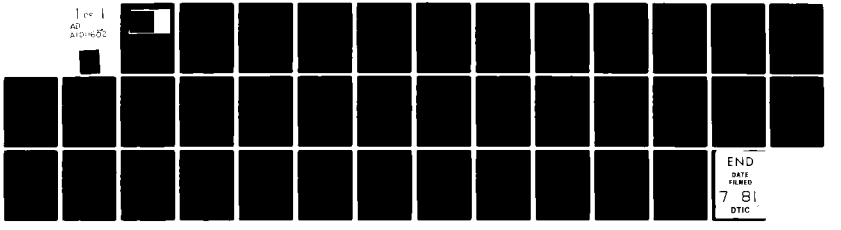
AD-A100 602 WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
ON PARAMETRIC LINEAR AND QUADRATIC PROGRAMMING PROBLEMS.(U)
MAR 81 K RITTER DAAG29-80-C-0041
UNCLASSIFIED MRC-TSR-2197 NL



LEVEL

MRC Technical Summary Report #2197

ON PARAMETRIC LINEAR AND QUADRATIC PROGRAMMING PROBLEMS

Klaus Ritter

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53706

Mar 81

(Received February 25, 1981)

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

ON PARAMETRIC LINEAR AND QUADRATIC PROGRAMMING PROBLEMS

Klaus Ritter

Technical Summary Report #2197

March 1981

ABSTRACT

An algorithm is described for determining the optimal solution of parametric linear and quadratic programming problems as an explicit piecewise linear function of the parameter. Each linear function is uniquely determined by an appropriate subset of active constraints. For every critical value of the parameter a new subset has to be determined. A simple rule is given for adding and deleting constraints from this subset.

AMS (MOS) Subject Classification: 90C31

Key Words: Linear programming, Quadratic programming, Parametric programming

Work Unit Number 5 - Operations Research

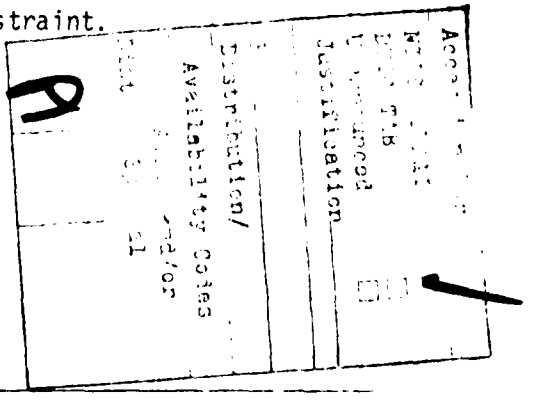
Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

SIGNIFICANCE AND EXPLANATION

In many applications of linear and quadratic programming it is necessary to obtain an optimal solution for more than one set of input data. In particular, if the right hand sides of the constraints $Ax \leq b$ can be interpreted as capacities, it might be useful to study the behavior of the optimal solution if b is replaced by $c + tp$, where t is a parameter which varies in some intervall, say $[\underline{t}, \bar{t}]$. Similarly, if the coefficients of the objective function $c'x$ are prices, it is sometimes desired to compute the optimal solution for all objective functions of the form $(c + tq)'x$ for $t \in [\underline{t}, \bar{t}]$.

Parametric problems of this type have the following basic property. There are critical values $\underline{t} = t_0 < t_1 < \ldots < t_\nu = \bar{t}$ such that the optimal solution is a linear function of t for $t_j \leq t \leq t_{j+1}$. This linear function can be computed from a linear system of equations which is determined by a certain set of active constraints. For each critical value of t this set of active constraints changes.

It can happen that for some critical values of t the new set of active constraints differs from the previous one by several constraints. In this case determination of the correct set can be tedious. This difficulty can be overcome by a simple selection rule which results in a finite number of intermediate sets which differ from each other by exactly one constraint. Therefore, the problem reduces to the simple case of consecutive sets of active constraints which are obtained from each other by adding or deleting or exchanging exactly one constraint.



ON PARAMETRIC LINEAR AND QUADRATIC PROGRAMMING PROBLEMS

Klaus Ritter

1. Introduction

In practice it is often important to study the behaviour of the optimal solution of a linear or nonlinear programming problem if some of the data change. In this paper we consider linear und convex quadratic minimization problems with the property that the right hand side of the constraints and/or the linear part of the objective function depend linearly on a parameter t which varies in a certain intervall. It is known [3] that in these cases the optimal solution is a piecewise linear function of the parameter. There is a finite number of critical values of the parameter for which the representation of the optimal solution as a function of t changes. These critical values are characterized by the fact that the set of constraints, active at the optimal solution changes. In a regular case when exactly one active constraint becomes inactive or exactly one inactive constraint becomes active it is not difficult to find the new representation of the optimal solution as a function of t. However, in a degenerate situation where for a critical value t_j of the parameter several constraints become newly active and/or several multipliers become zero, it may be tedious to find the correct set of constraints which determine the optimal solution for $t > t_j$.

It is the purpose of this paper to describe a method which can be used to overcome these difficulties. For every value of the parameter for which

---

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

the given problem has an optimal solution, the Kuhn-Tucker-conditions (see e.g. [2]) and an appropriately chosen set of constraints can be used to determine the optimal as a function of the parameter by solving a linear system of equations. For a regular critical value of t the selected set of constraints is changed in one of the following ways. Either a constraint is added to the set or a constraint is deleted from the set or both. In a degenerate case it could be necessary to repeat this procedure several times before the correct set of constraints is obtained. In order to avoid cycling the selection of the constraint that enters or leaves the considered set has to be made with some caution. It is shown that cycling does not occur if the following rules are used. First, adding a constraint to the selected set has priority over deleting a constraint from the set. Second, if there are several candidates for entering or leaving the set in each case the one with the smallest index will be chosen. The advantage of this method is that regular and degenerated critical values of t are treated in the same way. The only difference is that in the degenerate case more than one change of the selected set of constraints could be necessary.

In the following section we give a precise statement of the problem and establish some preliminary results. In Section 3 basic properties of parametric quadratic programming problems and their relation to the proposed method are given. In the final section an algorithm is described for computing the optimal solution as an explicit function of the parameter. It is shown that it terminates after a finite number of iterations.

2. Formulation of the problem and preliminary results

Let c, q, and x be n-dimensional column vectors and let A and b with

$$A' = (a_1,\ldots,a_m) \ , \quad b' = ((b)_1,\ldots,(b)_m)$$

be an (m,n)-matrix and an m-dimensional column vector, respectively. Furthermore, assume that C is a symmetric positive semi-definite (n,n)-matrix. We consider the problem of determining the optimal solution to

$$\min \left\{ (c+tq)'x + \frac{1}{2} x'Cx \mid Ax \leq b + tp \right\} \qquad (2.1)$$

as an explicit function of the parameter t for all values of t with

$$\underline{t} \leq t \leq \bar{t} .$$

If $C = 0$, then the parametric quadratic programming problem (2.1) reduces to a parametric linear programming problem.

Because C is assumed to be positive semidefinite it follows from the Kuhn-Tucker-Theorem (see e.g. [2]), that x is an optimal solution to (2.1) if and only if there is a vector $u \in E^m$ such that

$$\begin{aligned} Cx + A'u &= -(c + tq) \\ Ax &\leq b + tp \\ u'(Ax - b - tp) &= 0 , \quad u \geq 0 . \end{aligned} \qquad (2.2)$$

Throughout this paper we assume that

$$\text{rank } (C, A') = n . \qquad (2.3)$$

The purpose of this assumption is to guarantee that if (2.1) has an optimal solution for some t it also has an optimal solution, $\hat{x}$ say, which is the unique optimal solution in the intersection of the constraints that are active at $\hat{x}$. This fact is established in the following lemma.

Lemma 1

If rank $(C, A') = n$ and (2.1) has an optimal solution for some $\hat{t}$, then there is a set $I \subset \{1,2,\ldots,m\}$ and an optimal solution $\hat{x}$ of (2.1) such that

i) The vectors a_i, $i \in I$, are linearly independent

ii) $\hat{x}$ is the unique optimal solution to the problem

$$\min \left\{ (c + \hat{t}q)'x + \frac{1}{2} x'Cx \mid a_i' x = (b)_i + \hat{t}(p)_i , \ i \in I \right\} .$$

Proof:

For $t = \hat{t}$ let $x_0 = x_0(t)$ be any optimal solution to (2.1). Define the set $I_0 \subset \{1,2,\ldots,m\}$ such that $i \in I_0$ if and only if

$$a_i' x_o = (b)_i + \hat{t}(p)_i .$$

If x_o is the unique optimal solution to the problem

$$\min\left\{ (c + \hat{t}q)'x + \frac{1}{2} x'Cx \mid a_i' x = (b)_i + \hat{t}(p)_i , \ i \in I_o \right\} \tag{2.4}$$

it suffices to set $x = \hat{x}_o$ and to choose any maximal subset $I \subset I_o$ such that the vectors a_i, $i \in I$, are linearly independent.

Now suppose that $\hat{x}_o \neq x_o$ is an optimal solution to (2.1). Set $s = x_o - \hat{x}_o$ and

$$Q(x;\hat{t}) = (c + \hat{t}q)'x + \frac{1}{2} x'Cx .$$

Then it follows from the convexity of $Q(x; \hat{t})$ that

$$Q(x_o + \sigma s;\hat{t}) = Q(x_o;\hat{t}) \text{ for all } \sigma . \tag{2.5}$$

This implies that $s'Cs = 0$ and, because C is positive-semidefinite, $Cs = 0$. Therefore, it follows from (2.3) and the relations

$$a_i' s = 0 , \quad i \in I_o \tag{2.6}$$

that there is i, $i \notin I_o$, such that $a_i' s \neq 0$. Hence we deduce from (2.5) and (2.6) that, for some σ_o,

$$\hat{x}_1 = \hat{x}_o + \sigma_o s$$

is an optimal solution to (2.1) with

$$a_i' \hat{x}_1 = (b)_i + \hat{t}(p)_i , \quad i \in I_1 ,$$

where $I_o \subset I_1$ and $I_o \neq I_1$. Repeating this argument if necessary we obtain an optimal solution $\hat{x}_\nu$ to (2.1) and a set $I_\nu \subset \{1,2,\ldots,m\}$ such that $\hat{x}$ is the only optimal solution to (2.1) which satisfies the equations

$$a_i' x = (b)_i + \hat{t}(p)_i , \quad i \in I_\nu . \tag{2.7}$$

Setting $\hat{x} = x_\nu$ and choosing any maximal subset $I \subset I_\nu$ such that the vectors a_i, $i \in I$, are linearly independent completes the proof of the lemma.

In the following sections we will associate with each $\hat{t} \in [\underline{t}, \bar{t}]$ for which (2.1) has an optimal solution a matrix $\hat{A}$ and an optimal solution $\hat{x}$ with the properties specified in the above lemma. Let

$$a_\nu' x \leq (b)_\nu + t(p)_\nu \tag{2.8}$$

be any constraint which for $t = \hat{t}$ is active at the optimal solution $\hat{x}$. Then this constraint is said to be a primary active constraint if a_ν is a column of $\hat{A}'$. If a_ν is not a column of $\hat{A}'$, then (2.4) is referred to as a secondary active constraint.

For later reference we prove the following lemma.

Lemma 2

The matrix

$$\hat{M} = \begin{pmatrix} C & , & \hat{A}' \\ \hat{A} & , & 0 \end{pmatrix}$$

is nonsingular if and only if the following conditions are satisfied

i) The columns of $\hat{A}'$ are linearly independent.

ii) $x'Cx > 0$ for every $x \neq 0$ with $\hat{A}x = 0$.

Proof:

First assume that the conditions of the lemma are satisfied. Let $(\hat{x}, \hat{y})$ be any solution of the equations

$$\begin{aligned} Cx + \hat{A}'y &= 0 \\ \hat{A}x &= 0 . \end{aligned} \tag{2.9}$$

Then $\hat{x}'C\hat{x} = -\hat{y}'\hat{A}\hat{x} = 0$ implies $\hat{x} = 0$. Thus $\hat{A}'\hat{y} = 0$ and, therefore, $\hat{y} = 0$. This shows that $\hat{M}$ is nonsingular.

If the columns of $\hat{A}'$ are linearly dependent it follows immediately that $\hat{M}$ is singular. Finally, if there is $\dot{x} \neq 0$ with $\dot{x}'C\dot{x} = 0$ and $\hat{A}\dot{x} = 0$, then $C\dot{x} = 0$ and $(\dot{x}, 0)$ is a solution of the equations. Hence, $\hat{M}$ is singular.

3. Basic properties of parametric quadratic and linear programming problems

For $t = t_j$ let $x_j = x_j(t)$ be an optimal solution to (2.1) for which assumption (2.3) is satisfied. Re-numbering constraints if necessary and using Lemma 1 we can assume that

$$a_i' x_j = (b)_i + t_j(p)_i , \quad i=1,\ldots,\rho$$

$$a_i' x_j < (b)_i + t_j(p)_i , \quad i=\rho+1,\ldots,m$$

and x_j is the unique optimal solution to

$$\min \left\{ (c + t_j q)' x + \frac{1}{2} x' C x \mid A_j x = b_i + t_j p_j \right\} ,$$

where, for some $\nu \leq \rho$,

$$A_j' = (a_1,\ldots,a_\nu) , \quad b_j' = ((b)_1,\ldots,(b)_\nu) , \quad p_j' = ((p)_1,\ldots,(p)_\nu)$$

and the columns of A_j' are linearly independent.

Let $u_j = u_j(t_j) \geq 0$ be such that

$$C x_j(t_j) + A' u_j(t_j) = - (c + t_j q)$$

with $(u_j)_i = 0$ for $i=\nu+1,\ldots,m$. Then the optimality conditions (2.2) can be written in the form

$$\begin{aligned} C x_j(t) + A_j' v_j(t) &= - (c + t q) \\ A_j x_j(t) &= b_j + t p_j , \end{aligned} \tag{3.1}$$

$$a_i' x_j(t) \leq (b)_i + t(p)_i , \quad i=\nu+1,\ldots,m \tag{3.2}$$

$$v_j(t) \geq 0 , \tag{3.3}$$

where

$$v_j(t) \in E^\nu \text{ and } (v_j(t))_i = (u_j(t))_i , \quad i=1,\ldots,\nu .$$

Since by Lemma 2,

$$M_j = \begin{pmatrix} C , & A_j' \\ A_j , & 0 \end{pmatrix}$$

is nonsingular, the inverse matrix

$$M_j^{-1} = \begin{pmatrix} M_{1j} , & M_{2j} \\ M_{3j} , & M_{4j} \end{pmatrix}$$

exists.

Setting

$$h_{1j} = - M_{1j}(c + t_j q) + M_{2j}(b_j + t_j p_j) , \quad h_{2j} = - M_{1j} q + M_{2j} p_j$$

$$g_{1j} = - M_{3j}(c + t_j q) + M_{4j}(b_j + t_j p_j) , \quad g_{2j} = - M_{3j} q + M_{4j} p_j$$

we obtain from (3.1) the relations

$$x_j(t) = - M_{1j}(c + tq) + M_{2j}(b_j + t p_j) \tag{3.4}$$

$$= h_{1j} + (t - t_j) h_{2j}$$

and

$$v_j(t) = - M_{3j}(c + tq) + M_{4j}(b_j + t p_j) \tag{3.5}$$

$$= g_{1j} + (t - t_j) g_{2j} .$$

Substituting $x_j(t)$ into (3.2) we have, for $i=\nu+1,\ldots,m$, the inequality

$$a_i' h_{1j} + (t - t_j) a_i' h_{2j} \leq (b)_i + t_j (p)_i + (t - t_j)(p)_i \tag{3.6}$$

or

$$(t - t_j)(a_i' h_{2j} - (p)_i) \leq (b)_i + t_j (p)_i - a_i' h_{1j} .$$

Therefore, $x_j(t)$ is a feasible solution for all $t \geq 0$ for which these inequalities are satisfied.

Because $h_{1j} = x_j(t_j)$, it follows from (3.2) that

$$(b)_i + t_j(p)_i - a_i' h_{1j} \geq 0 , \quad i=\nu+1,\ldots,m \tag{3.7}$$

If $(a_i' h_{2j} - (p)_i) \leq 0$, $i=\nu+1,\ldots,m$, set $t_{j+1}^* = \infty$, otherwise set

$$t_{j+1}^* - t_j = \min \left\{ \frac{(b)_i + t_j(p)_i - a_i' h_{1j}}{a_i' h_{2j} - (p)_i} \;\middle|\; a_i' h_{2j} - (p)_i > 0 \right\} \tag{3.8}$$

and let k be the smallest index for which the minimum is attained.

Because of (3.7) we have $t_{j+1}^* \geq t_j$. Define the set I_{1j} of critical indices as follows

$$I_{1j} = \left\{ i \,\middle|\, (b)_i + t_j(p)_i - a_i' h_{1j} = 0 \text{ and } a_i' h_{2j} - (p)_i > 0 \right\} .$$

Clearly, $t_{j+1}^* > t_j$ if and only if $I_{1j} = \emptyset$.

If $I_{1j} \neq \emptyset$, then $x_j(t)$, as defined by (3.4), is not feasible for any $t > t_j$. Thus at least one of the secondary active constraints must become a primary active constraint. We choose the constraint with index

$$k = \min \left\{ i \,\middle|\, i \in I_{1j} \right\} . \tag{3.9}$$

The computation of the new matrix M_{j+1} is discussed below.

Suppose now that $I_{1j} = \emptyset$ and observe that by (3.3) we have

$$v_j(t_j) = g_{1j} \geq 0 . \tag{3.10}$$

If $g_{2j} \geq 0$, then $v_j(t) \geq 0$ for all $t \geq t_j$. In this case we set $\tilde{t}_{j+1} = \infty$; otherwise set

$$\tilde{t}_{j+1} - t_j = \min \left\{ \frac{(g_{1j})_i}{-(g_{2j})_i} \;\middle|\; (g_{2j})_i < 0 \right\} \tag{3.11}$$

and let l be the smallest index for which the minimum is attained. Because of (3.10), $\tilde{t}_{j+1} \geq t_j$. With

$$I_{2j} = \left\{ i \,\middle|\, (g_{1j})_i = 0 \text{ and } (g_{2j})_i < 0 \right\}$$

it follows immediately that $\tilde{t}_{j+1} > t_j$ if and only if $I_{2j} = \emptyset$.

If $I_{2j} \neq \emptyset$, then $x_j(t)$ is not optimal for any $t > t_j$. Thus at least one of the primary active constraints must become a secondary active constraint. We choose the constraint with index

$$l = \min \left\{ i \mid i \in I_{2j} \right\} . \tag{3.12}$$

The new matrix M_{j+1} is derived below.

Set $I_{2j} = \emptyset$. In order for $x_j(t)$ to be an optimal solution to (2.1), both the inequalities (3.6) and (3.5) have to be satisfied. Thus we set

$$t_{j+1} = \min \left\{ t^*_{j+1} , \tilde{t}_{j+1} \right\} .$$

Then the following lemma holds.

Lemma 3

There is $t_{j+1} \geq t_j$ such that

i) $x_j(t) = h_{1j} + (t - t_j) h_{2j}$ is an optimal solution to (2.1) for all t with $t_j \leq t \leq t_{j+1}$.

ii) There is no $\hat{t} > t_{j+1}$ such that $x_j(t)$ is an optimal solution to (2.1) for $t = \hat{t}$.

iii) $t_{j+1} > t_j$ if and only if $I_{1j} \cup I_{2j} = \emptyset$.

We now discuss the computation of the new matrix M_{j+1}.

Case 1:

Either $I_{1j} \neq \emptyset$ or $I_{1j} = I_{2j} = \emptyset$ and $t_{j+1} = t^*_{j+1}$. In either case the constraint

$$a_k' x \leq (b)_k + t(p)_k \tag{3.13}$$

has to be added to the set of primary active constraints. There are two cases depending on whether a_k is an element of span $\{a_1, \ldots, a_n\}$ or not. In order to decide which case applies we consider the equations

$$C w + A_j' z = a_k$$
$$A_i w = 0 ,$$

from which we obtain

$$w = M_{1j} a_k , \quad z = M_3 a_k . \tag{3.14}$$

Clearly, $a_k \in \text{span}\{a_1,\ldots,a_\nu\}$ if and only if $w = 0$.

First assume that $a_k \notin \text{span}\{a_1,\ldots,a_\nu\}$. In this case we set

$$A_{j+1}' = (A_j' , a_k)$$

and

$$M_{j+1} = \begin{pmatrix} C & , A_{j+1}' \\ A_{j+1} & , 0 \end{pmatrix} .$$

Since the columns of A_{j+1}' are linearly independent it follows from Lemma 2 that M_{j+1} is nonsingular.

Next assume that $w = 0$. Since in this case the vectors $a_1,\ldots,a_\nu,a_k$ are linearly dependent we have to determine a constraint

$$a_l' x \le (b)_l + t(p)_l$$

in the set of primary active constraints which will be replaced by (3.13). This can be done by using the vector z as defined by (3.14). If $z \le 0$, it follows from Lemma 6 in the next section that (2.1) has no feasible solution for $t > t_j$. If z has at least one positive component let the set I_{3j} be defined such that $i \in I_{3j}$ if and only if

$$\frac{(q_{1j})_i}{(z)_i} = \min \left\{ \frac{(q_{1j})_\nu}{(z)_\nu} \;\middle|\; (z)_\nu > 0 \right\}$$

and set

$$l = \min \left\{ i \;\middle|\; i \in I_{3j} \right\}$$

It is not difficult to verify that with this choice of l the gradients of the new set of primary active constraints are linearly independent and that there are $\lambda_i \ge 0$ such that

$$- (c + t_j q - C x_j(t_j)) = \sum_{\substack{i=1 \\ i \neq l}}^{\nu} \lambda_i a_i + \lambda_k a_k .$$

Furthermore, if $I_{1j} \neq \emptyset$ it follows from Lemma 7 that the above rules for choosing k and l ensure the existence of an $j_1 > j$ such that $I_{1j_1} = \emptyset$.

Case 2:

Either $I_{1j} = \emptyset$, $I_{2j} \neq \emptyset$ or $I_{1j} = I_{2j} = \emptyset$ and $t_{j+1} = \tilde{t}_{j+1} < t^*_{j+1}$. In either case the constraint

$$a_l' x \leq (b)_l + t(q)_l$$

has to be deleted from the set of primary active constraints. If C is not positive definite in the set

$$T = \left\{ x \,\middle|\, a_i' x = 0 , \quad i=1,\ldots,\nu , \quad i \neq l \right\} \tag{3.15}$$

or if there are secondary active constraints at $x_j(t_j)$ it may be necessary to add a constraint

$$a_k' x \leq (b)_k + t(q)_k$$

to the set of primary active constraints. In order to decide this we denote the l-th column of M_{2j} by s_l and the element in the l-th row and l-th column of M_{4j} by ω. Then it follows from Lemma 4 that C is positive definite in the set (3.15) if and only if $\omega < 0$. Furthermore, if $\omega < 0$, then

$$x_j(t) = \sigma_o(t - t_j)s_l = h_{1j} + (t - t_j)(h_{2j} - \sigma_o s_l) \tag{3.16}$$

is the unique optimal solution to the problem

$$\min \left\{ (c + tq)'x + \frac{1}{2} x'C x \;\middle|\; a_i' x = (b)_i + t(p)_i , \quad i=1,\ldots,\nu , \quad i \neq l \right\},$$

where $\sigma_o = \dfrac{(q_{2j})_l}{\omega}$ and q_{2j} is the vector used in (3.5). If there are secondary active constraints at $x_j(t_j)$, (3.16) need not be a feasible solution to (2.1) for any $t > t_j$. Therefore, we define the set

$$I_{4j} = \left\{ i \,\middle|\, a_i' s_l < 0 \text{ and } a_i' h_{1j} = (b)_i + t_j(p)_i \right\}$$

and, if $I_{4j} \neq \emptyset$, the number

$$\tau_o = \min \left\{ \frac{(p)_i - a_i' h_{2j}}{-a_i' s_1} \;\middle|\; i \in I_{4j} \right\}.$$

If $\sigma_o \leq \tau_o$, then (3.16) will be a feasible solution to (2.1) for $t > t_j$ sufficiently small.

Therefore, if $\omega < 0$ and $\sigma_o < \tau_o$ no new primary active constraint is needed. If $\omega < 0$ and $\sigma_o \geq \tau_o$, then a secondary active constraint will become a primary active constraint. Finally, if $\omega \geq 0$ and $I_{4j} = \emptyset$ then the problem

$$\min \left\{ (c + tq)'x + \frac{1}{2} x'C x \;\middle|\; a_i' x = (b)_i + t(p_i),\ i=1,\ldots,\nu,\ i \neq 1 \right\}$$

has no optimal solution for $t > t_j$ (see the proof of Lemma 6). In this case either (2.1) has no optimal solution for $t > t_j$ or there is some $\tau_1 > 0$ and some index k such that $a_k' s_1 < 0$,

$$x_j(t_j) - \tau_1 s_1$$

is an optimal solution to (2.1) and $a_k' x_j(t_j) - \tau_1 a_k' s_1 = (b)_k + t_j(p)_k$, where τ_1 and k are determined by inserting $x_j(t_j) - \tau s_1$ into the inequalities

$$a_i' x \leq (b)_i + t(p)_i, \quad i=\nu+1,\ldots,m$$

which are not active at $x_j(t_j)$ and computing the largest value of τ for which $x_j(t_j) - \tau s_1$ is feasible. Details of this procedure are given in Step 3.3 of the algorithm described in the next section.

It follows from Lemma 9 that the above rules for changing the set of primary active constraints ensure the existence of an $j_1 > j$ such that $I_{1\nu} = \emptyset$, $\nu = j, j+1, \ldots, j_1$, and $I_{2\nu} = \emptyset$.

4. An algorithm for solving parametric linear and quadratic programming problems

In this section we describe an algorithm for computing the optimal solution to problem (2.1) as an explicit function of the parameter t. It is assumed that, for $t = \underline{t}$, the algorithm starts with an optimal solution $x_o(\underline{t})$ to problem (2.1) which has the properties described in Lemma 2.1.

A general cycle of the algorithm consists of four steps. At the beginning of the j-th cycle the following data are available:

$$M_j^{-1} = \begin{pmatrix} M_{1j} , M_{2j} \\ M_{3j} , M_{4j} \end{pmatrix}, \quad h_{1j} , \quad h_{2j} , \quad g_{1j} , \quad g_{2j} , \quad J(x_j) \text{ and } \nu_j .$$

Here ν_j denotes the number of primary active constraints. The elements α_{ij} of the index set $J(x_j)$ are positive integers and are defined in such a way that $\alpha_{ij} = k$ if and only if the i-th column of the matrix A_j' is equal to a_k, the gradient of a primary active constraint. Then

$$M_j = \begin{pmatrix} C , A_j' \\ A_j , 0 \end{pmatrix}$$

and

$$\begin{pmatrix} h_{1j} \\ g_{1j} \end{pmatrix} = M_j^{-1} \begin{pmatrix} - c - t_j q \\ b_j + t_j p_j \end{pmatrix}$$

$$\begin{pmatrix} h_{2j} \\ g_{2j} \end{pmatrix} = M_j^{-1} \begin{pmatrix} - q \\ p_j \end{pmatrix} ,$$

where b_j and p_j are appropriate subvectors of b and p, respectively.

In Step 1.1 and Step 1.2 the critical sets I_{1j} and I_{2j} are determined which contain the indices of secondary active constraints which are not satisfied for $t > t_j$ and multipliers which are negative for $t > t_j$, respectively. If both sets are empty Step 3.1 is used to compute the maximal $t_{j+1} > t_j$ such that

$$x_j(t) = h_{1j} + (t - t_j) h_{2j}$$

is an optimal solution to problem (2.1) for $t_j \le t \le t_{j+1}$.

In Step 2 it is determined whether a_k, the gradient of the constraint that becomes a new primary active constraint, is linearly dependent on the gradients of the present set of primary active constraints. If this is the case then a constraint is selected which has to be deleted from the set of primary active constraints.

In Step 3 a constraint $a_l' x \le (b)_l + t(p)_l$ is given which will be dropped from the set of primary active constraints. If the problem

$$\min \left\{ (c + tq)'x + \frac{1}{2} x'Cx \;\middle|\; a_i' x = (b)_i + t(p)_i,\ i \in J(x_j) - \{l\} \right\}$$

has a unique optimal solution which satisfies the inequalities $Ax \le b + tp$ for $t > t_{j+1}$ sufficiently small, then no new primary active constraint is required; otherwise a constraint is determined that will be added to the set of primary active constraints.

In Step 4, the matrices A_{j+1} and M_{j+1}^{-1} are computed.

Next we give a detailed description of a general cycle of the algorithm.

Step 1

1.1 Define the set I_{1j} such that $i \in I_{1j}$ if and only if

$$(b)_i + t_j(p)_i - a_i' h_{1j} = 0 \quad \text{and} \quad a_i' h_{2j} - (p)_i > 0 .$$

If $I_{1j} = \emptyset$ go to Step 1.2; otherwise set

$$k = \min \left\{ i \;\middle|\; i \in I_{1j} \right\}, \quad t_{j+1} = t_j$$

and go to Step 2.

1.2 Define the set I_{2j} such that $i \in I_{2j}$ if and only if

$$(q_{1j})_i = 0 \quad \text{and} \quad (q_{2j})_i < 0 .$$

If $I_{2j} = \emptyset$ go to Step 1.3; otherwise set

$$l = \min \left\{ \alpha_{ij} \;\middle|\; i \in I_{2j} \right\}, \quad t_{j+1} = t_j$$

and go to Step 3.1.

1.3 If $a_i' h_{2j} - (p)_i \le 0$ for $i=1,\ldots,m$ set $t^*_{j+1} - t_j = \infty$, otherwise set

$$t^*_{j+1} - t_j = \min \left\{ \frac{(b)_i + t_j(p)_i - a_i' h_{1j}}{a_i' h_{2j} - (p)_i} \;\middle|\; a_i' h_{2j} - (p)_i > 0 \right\}$$

and let k be the smallest index for which the minimum is attained. If $g_{2j} \ge 0$ set $\tilde{t}_{j+1} - t_j = \infty$, otherwise set

$$\tilde{t}_{j+1} - t_j = \min \left\{ \frac{(g_{1j})_i}{-(g_{2j})_i} \;\middle|\; (g_{2j})_i < 0 \right\}$$

and let l be the smallest index for which the minimum is attained. Set

$$t_{j+1} = \min \left\{ t^*_{j+1}, \tilde{t}_{j+1} \right\}$$

and print

$$t_{j+1}, \quad h_{1j} \quad \text{and} \quad h_{2j}.$$

If $t_{j+1} \ge \bar{t}$ stop; otherwise do the following. If $t_{j+1} = t^*_{j+1}$ go to Step 2. If $t_{j+1} = \tilde{t}_{j+1} < t^*_{j+1}$ go to Step 3.1.

Step 2

Compute

$$w = M_{1j} a_k .$$

If $w \neq 0$ go to Step 4.1, otherwise compute

$$z = M_{3j} a_k .$$

If $z \le 0$ stop with the message that the problem (2.1) has no feasible solution for $t > t_{j+1}$. If z has at least one positive component define the set I_{3j} such that $i \in I_{3j}$ if and only if

$$\frac{(g_{1j})_i}{(z)_i} = \min \left\{ \frac{(g_{1j})_\nu}{(z)_\nu} \;\middle|\; (z)_\nu > 0 \right\} .$$

Set

$$l = \min \left\{ \nu_{ij} \;\middle|\; i \in I_{3j} \right\}$$

and go to Step 4.3.

Step 3

3.1 Set $s_1 = (M_{2j})_1$, $\omega = (M_{4j})_{11}$, and

$$I_{4j} = \left\{ i \,\middle|\, a_i' s_1 < 0 \text{ and } a_i' h_{1j} = (b)_i + t_j(p)_i \right\} .$$

If $I_{4j} = \emptyset$ and $\omega = 0$, go to Step 3.3.

If $I_{4j} = \emptyset$ and $\omega \neq 0$, go to Step 4.2.

If $I_{4j} \neq \emptyset$, compute

$$\tau_o = \min \left\{ \frac{(p)_i - a_i' h_{2j}}{- a_i' s_1} \,\middle|\, i \in I_{4j} \right\}$$

and let k be the smallest index for which the minimum is attained. If $\tau_o > 0$, go to Step 3.2; otherwise compute

$$w = M_{1j} a_k .$$

If $w \neq 0$, go to Step 4.1; otherwise go to Step 4.3.

3.2 If $\omega \neq 0$ and $\frac{(g_{2j})_1}{\omega} < \tau_o$, go to Step 4.2; otherwise go to Step 4.3.

3.3 Set

$$I_{5j} = \left\{ i \,\middle|\, a_i' s_1 < 0 \text{ and } a_i' h_{1j} < (b)_i + t_j(p)_i \right\} .$$

If $I_{5j} = \emptyset$, stop with the message that there is no optimal solution for $t > t_{j+1}$. If $I_{5j} \neq \emptyset$, compute

$$\tau_1 = \min \left\{ \frac{(b)_i + t_j(p)_i - a_i' h_{1j}}{- a_i' s_1} \,\middle|\, i \in I_{5j} \right\}$$

and set

$$I_{6j} = \left\{ i \in I_{5j} \,\middle|\, \tau_1 = \frac{(b)_i + t_j(p)_i - a_i' h_{1j}}{- a_i' s_1} \right\} .$$

Compute

$$\tau_2 = \min \left\{ \frac{(p)_i - a_i' h_{2j}}{- a_i' s_1} \,\middle|\, i \in I_{6j} \right\}$$

and let k be the smallest index for which the minimum is attained. Go to Step 4.3.

Step 4

4.1: Set $\nu_{j+1} = \nu_j + 1$ and $J_{j+1} = \{\alpha_{1,j+1},\ldots,\alpha_{\nu_{j+1},j+1}\}$, where

$$\alpha_{i,j+1} = \alpha_{ij} \ , \quad i=1,\ldots,\nu_j \ , \quad \alpha_{\nu_{j+1},j+1} = k \ .$$

Go to Step 4.4.

4.2: Set $\nu_{j+1} = \nu_j - 1$ and $J_{j+1} = \{\alpha_{1,j+1},\ldots,\alpha_{\nu_{j+1},j+1}\}$, where

$$\alpha_{i,j+1} = \alpha_{ij} \ , \quad i=1,\ldots,l-1$$

$$\alpha_{i,j+1} = \alpha_{i+1,j} \ , \quad i=l,\ldots,\nu_j - 1 \ .$$

Go to Step 4.4.

4.3: Set $\nu_{j+1} = \nu_j$ and $J_{j+1} = \{\alpha_{1,j+1},\ldots,\alpha_{\nu_{j+1},j+1}\}$, where

$$\alpha_{i,j+1} = \alpha_{ij} \ , \quad i=1,\ldots,\nu_j \ , \quad i \neq l$$

$$\alpha_{l,j+1} = k \ .$$

Go to Step 4.4.

4.4: Set

$$A'_{j+1} = \left(a_{\alpha_{1,j+1}},\ldots,a_{\alpha_{\nu_{j+1},j+1}}\right)$$

$$b'_{j+1} = \left((b)_{\alpha_{1,j+1}},\ldots,(b)_{\alpha_{\nu_{j+1},j+1}}\right)$$

$$P'_{j+1} = \left((p)_{\alpha_{1,j+1}},\ldots,(p)_{\alpha_{\nu_{j+1},j+1}}\right)$$

$$M_{j+1} = \begin{pmatrix} C & , & A'_{j+1} \\ A_{j+1} & , & 0 \end{pmatrix} \qquad M^{-1}_{j+1} = \begin{pmatrix} M_{1,j+1} & , & M_{2,j+1} \\ M_{3,j+1} & , & M_{4,j+1} \end{pmatrix}$$

$$\begin{pmatrix} h_{1,j+1} \\ g_{2,j+1} \end{pmatrix} = \begin{pmatrix} M_{1,j+1} & M_{2,j+1} \\ M_{3,j+1} & M_{4,j+1} \end{pmatrix} \begin{pmatrix} -c - t_{j+1} q \\ b_{j+1} + t_{j+1} p_{j+1} \end{pmatrix}$$

$$\begin{pmatrix} h_{2,j+1} \\ g_{2,j+1} \end{pmatrix} = \begin{pmatrix} M_{1,j+1} & M_{2,j+1} \\ M_{3,j+1} & M_{4,j+1} \end{pmatrix} \begin{pmatrix} -q \\ p_{j+1} \end{pmatrix} .$$

Replace j with $j+1$ and go to Step 1.1.

In the following we prove some lemmas which establish the basic properties of the algorithm. The first two lemmas are concerned with the existence of M_j^{-1} .

Lemma 4

Let $s_1 = (M_{2j})_1$ and $\omega = (M_{4j})_{11}$ be defined as in Step 3.1 of the algorithm and assume that

$$x'Cx > 0 \quad \text{for} \quad x \neq 0 , \quad x \in \left\{ x \,\middle|\, a_i' x = 0 , \; i \in J_j \right\} .$$

i) Then

$$x'Cx > 0 \quad \text{for} \quad x \neq 0 , \quad x \in T = \left\{ x \,\middle|\, a_i' x = 0 , \; i \in J_j - \{1\} \right\}$$

if and only if $\omega < 0$.

ii) If $\omega < 0$ set $\sigma_0 = (g_{2j})_1 / \omega$. Then

$$x_j(t) - \sigma_0 (t - t_{j+1}) s_1$$

is the unique optimal solution to the problem

$$\min \left\{ (c + tq)'x + \tfrac{1}{2} x'Cx \,\middle|\, a_i' x = (b)_i + t(p)_i , \; i \in J_j - \{1\} \right\} . \quad (4.1)$$

Proof:

Set $s_2 = (M_{4j})_1$. Then

$$Cs_1 + A_i' s_2 = 0 , \quad a_i' s_1 = 0 , \quad i \in J_j - \{1\}$$

and

$$a_1' s_1 = 1 .$$

Thus

$$s_1' C s_1 = - s_1' A_j s_2 = - (s_2)_1 = - (M_{4j})_{11} = - \omega . \tag{4.2}$$

Let $x \in T$ and $x \neq 0$. Then there are λ and y such that $A_j' y = 0$ and $x = y + \lambda s_1$. Therefore,

$$\begin{aligned} x' C x &= y' C y + 2 \lambda s_1' C y + \lambda^2 s_1' C s_1 \\ &= y' C y + \lambda^2 s_1' C s_1 , \end{aligned} \tag{4.3}$$

because $s_1' C y = - s_2' A_j y = 0$. Since $s_1 \in T$, the first part of the lemma follows from (4.2) and (4.3).

In order to prove the second part of the lemma we first observe that

$$a_i'(x_j(t) - \sigma_o(t - t_{j+1})s_1) = (b)_i + t(p)_i , \quad i \in J_j - \{1\} .$$

Furthermore,

$$C x_j(t) + A_j' v_j(t) = - (c + tq) \tag{4.4}$$

with $(v_j(t_{j+1}))_1 = 0$. Thus

$$C[x_j(t) - \sigma_o(t - t_{j+1})s_1] + A_j'[v_j(t) + \sigma_o(t - t_{j+1})s_2] = - (c + tq) \tag{4.5}$$

and

$$\begin{aligned} (v_j(t) + \sigma_o(t - t_{j+1})s_2)_1 &= (v_j(t_{j+1}))_1 + [t - t_{j+1}](g_{2j} + \sigma_o s_2)_1 \\ &= [t - t_{j+1}]((g_{2j})_1 + \sigma_o(s_2)_1) = 0 . \end{aligned}$$

Therefore, it follows from (4.4), (4.5) and the Kuhn-Tucker-conditions that $x_j(t) - \sigma_o(t - t_{j+1})s_1$ is an optimal solution to (4.1). By the first part of the lemma it is the unique optimal solution.

Lemma 5

If M_o is a nonsingular matrix, then every matrix

$$M_j = \begin{pmatrix} C & , & A_j' \\ A_j & , & 0 \end{pmatrix}$$

determined by the algorithm is nonsingular.

Proof:
Suppose that, for some j, M_j is nonsingular. Because of Lemma 2 we know that then the columns of A_j' are linearly independent and $x'Cx > 0$ for every $x \neq 0$ with

$$x \in T_j = \left\{ x \mid A_j x = 0 \right\} .$$

Similarly, it follows from Lemma 2 that M_{j+1} is nonsingular if the columns of A_{j+1}' are linearly independent and $x'Cx > 0$ for every $x \neq 0$ with

$$x \in T_{j+1} = \left\{ x \mid A_{j+1} x = 0 \right\} .$$

Without loss of generality we may assume that

$$A_j' = (a_1, \ldots, a_\nu) .$$

First assume that Step 2 of the algorithm is used in the j-th cycle. Then

$$\begin{pmatrix} w \\ z \end{pmatrix} = \begin{pmatrix} M_{1j} , M_{2j} \\ M_{3j} , M_{4j} \end{pmatrix} \begin{pmatrix} a_k \\ 0 \end{pmatrix} = \begin{pmatrix} M_{1j} & a_k \\ M_{3j} & a_k \end{pmatrix}$$

and

$$Cw + A_j' z = a_k .$$

If $w \neq 0$, then $A_{j+1}' = (a_1, \ldots, a_\nu, a_k)$.
If $w = 0$, then $A_{j+1}' = (a_1, \ldots, a_{l-1}, a_k, a_{l+1}, \ldots, a_\nu)$ with $(z)_l \neq 0$.
In both cases it follows immediately that the columns of A_{j+1}' are linearly independent. Furthermore, $x' C x > 0$ for every $x \neq 0$ with $x \in T_{j+1}$ because $T_{j+1} \subseteq T_j$.

Next assume that Step 3 of the algorithm is used in the j-th cycle. If

$$A_{j+1}' = (a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_\nu)$$

then $\omega \neq 0$. The columns of A_{j+1}' are linearly independent and it follows from Lemma 4 that $x' C x > 0$ for $x \neq 0$, $x \in T_{j+1}$. If

$$A_{j+1}' = (a_1, \ldots, a_{l-1}, a_k, a_{l+1}, \ldots, a_\nu)$$

then $a_k' s_l < 0$.

Since $a_i' s_1 = 0$, $i=1,\ldots,\nu$, $i \neq 1$, the columns of A_{j+1}' are linearly independent. Furthermore,

$$x \in \hat{T}_j = \left\{ x \,\middle|\, a_i' x = 0 \ , \ i=1,\ldots,\nu \ , \ i \neq 1 \right\}$$

and $x' C x = 0$ imply $x = \lambda s_1$ for some λ . Since $T_{j+1} = \hat{T}_j \cap \{x | a_k' x = 0\}$ and $a_k' s_1 < 0$ the vector s_1 is not an element of T_{j+1} . Thus $x \in T_{j+1}$ and $x \neq 0$ imply $x' C x > 0$.

The next lemma shows that the termination of the algorithm with Step 2 or 3 implies that the given problem has no optimal solution for $t > t_{j+1}$.

Lemma 6

i) If the algorithm terminates with Step 2 in the j-th cycle, then the problem (2.1) has no feasible solution for any $t > t_{j+1}$.

ii) If the algorithm terminates with Step 3 in the j-th cycle, then the problem (2.1) has no optimal solution for any $t > t_{j+1}$.

Proof:

i) We have $a_k = A_j' z$, $z \leq 0$ and, by the definition of k ,

$$a_k' h_{1j} = (b)_k + t_{j+1}(p)_k \ , \quad a_k' h_{2j} > (p)_k \ .$$

Let $t > t_{j+1}$ and $A_j x \leq b_j + t_j p_j + (t - t_j) p_j$. Then

$$A_j(x - h_{1j} - (t - t_j) h_{2j}) \leq 0$$

and

$$a_k'(h_{1j} - (t - t_j) h_{2j}) > (b)_k + t_j (p)_k + (t - t_j)(p)_k \ .$$

Therefore,

$$\begin{aligned} a_k' x &= a_k'(h_{1j} - (t - t_j)h_{2j}) + z' A_j(x - h_{1j} - (t - t_j)h_{2j}) \\ &> (b)_k + t_j(p)_k + (t - t_j)(p)_k \ . \end{aligned}$$

ii) Without loss of generality we may assume that $A_j' = (a_1,\ldots,a_\nu)$ and $l = \nu$. By the definition of l we have

$$(v_j(t_{j+1}))_l = 0 \quad \text{and} \quad (q_{2j})_l < 0 \ .$$

Furthermore, let s_1 and s_2 be defined as in the proof of Lemma 4. Then

$$s_1' C s_1 = -\omega = 0 .$$

Because C is positive semidefinite this implies $C s_1 = 0$. Thus for any $\hat{t} > t_{j+1}$, we have

$$s_1' C x_j(\hat{t}) + s_1' A_j v_j(\hat{t}) = - s_1'(c + \hat{t} q)$$

or

$$- s_1'(c + \hat{t} q) = (v_j(\hat{t}))_1 = (g_{1j})_1 + (\hat{t} - t_{j+1})(g_{2j})_1 < 0 .$$

Furthermore, since $I_{4j} = \emptyset$ and $I_{5j} = \emptyset$ it follows that $a_i' s_1 \geq 0$ for $i=1,\ldots,m$. Therefore, if for any $\hat{t} > t_{j+1}$, there is a feasible solution $\hat{x}$, say, then

$$a_i'(\hat{x} - \sigma s_1) \leq (b)_i + \hat{t}(p)_i \quad \text{for all} \quad \sigma \geq 0$$

and $i=1,\ldots,m$. Moreover,

$$(c + \hat{t} q)'(\hat{x} - \sigma s_1) + \frac{1}{2} (\hat{x} - \sigma s_1)' C(\hat{x} - \sigma s_1) =$$

$$(c + \hat{t} q)' \hat{x} + \frac{1}{2} \hat{x}' C \hat{x} - \sigma (c + \hat{t} q)' s_1 \to -\infty \quad \text{as} \quad \sigma \to \infty$$

The main difficulty in showing the finite termination of the algorithm is to prove that for every j for which the union of the critical sets I_{1j} and I_{2j} is not the empty set there is some $j_1 > 0$ such that $I_{1j_1} \cup I_{2j_1} = \emptyset$. This is done in the following three lemmas.

Lemma 7

For every j_o with $I_{1j_o} \neq \emptyset$ there is $j_1 > j_o$ with $I_{1j_1} = \emptyset$.

Proof:

There is a largest integer $j^* \geq j_o$ with

$$I_{1j} \neq \emptyset , \quad j=j_o, j_o+1, \ldots, j^*$$

and

$$a_k \notin \text{span} \left\{ a_i \mid i \in J(x_j^*) \right\} ,$$

where k is the index determined in Step 1.1 of the algorithm. Set $j = j^* + 1$

and let D be any matrix such that (A_j', D') is a nonsingular (n,n)-matrix and every column of D' is orthogonal to all columns of A_j'. Define the set of integers I in such a way that $i \in I$ if and only if

$$a_i' x_j(t_j) = (b)_i + t_j(p)_i \tag{4.6}$$

and

$$a_i \in \text{span} \left\{ a_i \mid i \in J(x_j) \right\} . \tag{4.7}$$

Finally choose any $t > t_j$ and consider the following linear programming problem

$$\max \left\{ (b_j + t p_j)' v + \sum_{i \in I} ((b)_i + t(p)_i) \lambda_i \right\}$$

subject to the constraints

$$A_j' v + D y + \sum_{i \in I} a_i \lambda_i = -(c + t_j q) - C x_j(t_j)$$

$$v \geq 0 , \quad \lambda_i \geq 0 , \quad i \in I .$$

We will apply the simplex method with Bland's [1] minimum index rule to this problem.

First we observe that (A_j', D) is a basis matrix for this problem. The corresponding basic solution is $v = v_j(t_j)$, $y = 0$. In order to determine the new basic variable we compute

$$\hat{x} = \begin{pmatrix} A_j' \\ D \end{pmatrix}^{-1} (b_j + t p_j)$$

and

$$\alpha_i = (b)_i + t(p)_i - a_i' \hat{x} , \quad i \in I .$$

If $\alpha_i \geq 0$ for all $i \in I$, then the current basic solution is optimal. If at least one α_i is negative, then λ_k will become the new basic variable, where

$$k = \min \left\{ i \in I \mid \alpha_i < 0 \right\} . \tag{4.8}$$

Because $\hat{x}$ and $x_j(t)$ are solutions to the equations $A_j x = b_j + t p_j$, we have $A_j(\hat{x} - x_j(t)) = 0$. Thus it follows from (4.7) that

$$\alpha_i = (b)_i + t(p)_i - a_i' x_j(t) , \quad i \in I . \tag{4.9}$$

Since $x_j(t) = h_{1j} + (t - t_j)h_{2j}$ we deduce from (4.6) and (4.9) that

$$a_i' h_{2j} - (p)_i > 0 \quad \text{for every} \quad i \in I \quad \text{with} \quad \alpha_i < 0 .$$

Therefore, every $i \in I$ with $\alpha_i < 0$ is an element of the set I_{1j}. Since by the definition of j^*, the index k determined in Step 1.1 of the algorithm is an element of the set I, it is identical with the k selected by (4.8).

In order to determine the basic variable that will leave the basis we compute

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = (A_j' , D')^{-1} a_k .$$

From (4.7) we deduce that $y_2 = 0$ and $y_1 = z$, where $z = M_{3j} a_k$ is the vector determined in Step 2.1 of the algorithm as part of the solution to the equations

$$\begin{aligned} C w + A_j' z &= a_k \\ A_j w &= 0 . \end{aligned}$$

Defining the set $\hat{I}$ such that $i \in \hat{I}$ if and only if

$$\frac{(v_j(t_j))_i}{(z)_i} = \min \left\{ \frac{(v_j(t_j))_\nu}{(z)_\nu} \;\middle|\; z_\nu > 0 \right\}$$

we obtain the index of the basic variable that will leave the basis as follows

$$l = \min \left\{ \alpha_{ij} \;\middle|\; i \in \hat{I} \right\} .$$

This index is the same as the one determined in Step 2 of the algorithm.

The above results show that, for $j = j^* + 1$, $I_{1j} = \emptyset$ if and only if $v = v_j(t_j)$, $y = 0$ is an optimal basic solution to a linear maximization problem. If $I_{1j} \neq \emptyset$, then (A_{j+1}' , D') is the basis matrix obtained by performing one iteration in the simplex method. The corresponding basic solution is optimal if and only if $I_{2,j+1} = \emptyset$. Repeating this argument and

observing that by Bland's rule the simplex methods determines on optimal solution in a finite number of iterations we deduce that there is $j_1 > j_o$ with $I_{1j_1} = \emptyset$.

Lemma 8

For every j , $I_{1j} = \emptyset$ and $I_{2j} \neq \emptyset$ imply $I_{1,j+1} = \emptyset$.

Proof:

Let $s_1 = (M_{2j})_1$ and $s_2 = (M_{4j})_1$. Then (s_1, s_2) satisfies the equations

$$\begin{aligned} C\, s_1 + A_j'\, s_2 &= 0 \\ A_j\, s_1 &= e_j \,, \end{aligned} \tag{4.10}$$

where $(e_j)_i = 0$, $i \neq 1$, and $(e_j)_1 = 1$.

Furthermore,

$$x_j(t) = h_{1j} + (t - t_j)h_{2j} \,, \quad v_j(t) = g_{1j} + (t - t_j)g_{2j}$$

are determined as the unique solution to the equations

$$\begin{aligned} C\, x + A_j'\, v &= -(c + t\,q) \\ A_j\, x &= b_j + t\, p_j \,. \end{aligned}$$

Without loss of generality we can assume that

$$A_j' = (a_1, a_2, \ldots, a_\nu) \quad \text{and} \quad 1 = \nu \,.$$

Then

$$x_{j+1}(t) = h_{1,j+1} + (t - t_{j+1})h_{2,j+1}$$

$$v_{j+1}(t) = g_{1,j+1} + (t - t_{j+1})g_{2,j+1}$$

is the unique solution to the equations

$$\begin{aligned} C\, x + A_{j+1}'\, v &= -(c + t\,q) \\ A_{j+1}\, x &= b_{j+1} + t\, p_{j+1} \,, \end{aligned} \tag{4.11}$$

where $A_{j+1}' = (a_1, \ldots, a_{\nu-1})$ or $A_{j+1}' = (a_1, \ldots a_\nu, a_k)$ or $A_{j+1} = (a_1, \ldots a_{\nu-1}, a_k)$, depending on which part of Step 3 of the algorithm applies. Because $I_{2j} \neq \emptyset$ we have $t_{j+1} = t_j$.

Case 1:

$I_{4j} \neq \emptyset$ and $\tau_o = 0$.

Since $a_k' h_{2j} - (p)_k = 0$, it follows that

$$a_k' x_j(t) = (b)_k + t(p)_k + (t - t_j)(p)_k \tag{4.12}$$

for all t .

If $M_{1j} a_k \neq 0$, then $A_{j+1}' = (A_j , a_k)$ and it follows from (4.12) that

$$x_j(t) , \ (v_j(t) , 0)$$

is a solution to (4.11). Thus $x_{j+1}(t) = x_j(t)$. If i is the index of a secondary active constraint, we have, therefore,

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i \leq 0 ,$$

i.e. $I_{1,j+1} = \emptyset$.

Now assume that $M_{1j} a_k = 0$. Then $A_{j+1}' = (a_1,\ldots,a_{\nu-1},a_k)$ and $a_k = A_j' z$ with $z = M_{2j} a_k$. Since $(z)_1 = z' A_j s_1 = a_k' s_1 \neq 0$, we can define

$$(\hat{g}_{1j})_i = (g_{1j})_i - \frac{(g_{1j})_1}{(z)_1} (z)_i , \quad i=1,\ldots,\nu-1$$

$$(\hat{g}_{1j})_1 = \frac{(g_{1j})_1}{(z)_1}$$

$$(\hat{g}_{2j})_i = (g_{2j})_i - \frac{(g_{2j})_1}{(z)_1} (z)_i , \quad i=1,\ldots,\nu-1$$

$$(\hat{g}_{2j})_1 = \frac{(g_{2j})_1}{(z)_1} .$$

Then

$$A_{j+1}' \hat{v}_j(t) = A_j' v_j(t) , \quad \hat{v}_j(t) = \hat{g}_{1j} + (t - t_j)\hat{g}_{2j} . \tag{4.13}$$

Therefore, it follows from (4.12) and (4.13) that $x_j(t)$, $\hat{v}_j(t)$ is a solution to (4.11). Thus

$$x_{j+1}(t) = x_j(t) \tag{4.14}$$

and $I_{1,j+1} = \emptyset$.

Case 2:

$(I_{4j} = \emptyset$ and $\omega \neq 0)$ or $(I_{4j} \neq \emptyset$, $\tau_o > 0$, $\omega \neq 0$, and $(g_{2j})_1 / \omega < \tau)$.

In this case we have $A'_{j+1} = (a_1,\ldots,a_{\nu-1})$. With

$$\sigma_o = \frac{(g_{2j})_1}{\omega} \text{ , } \omega = (M_{4j})_{11} = (s_2)_1$$

we have

$$(v_j(t))_1 - \sigma_o(t - t_j)(s_2)_1 = (t - t_j)((g_{2j})_1 - \sigma_o(s_2)_1) = 0 .$$

Let $\hat{v}_j(t)$ and $\hat{s}_2$ denote the vectors obtained from $v_j(t)$ and s_2 , respectively, by deleting the 1-th component. Then

$$x_j(t) - \sigma_o(t - t_j)s_1 \text{ , } \hat{v}_j(t) - \sigma_o(t - t_j)\hat{s}_2$$

satisfy the equations (4.11). Indeed,

$$A_{j+1}\, x_j(t) - \sigma_o(t - t_j)A_{j+1}\, s_1 = A_{j+1}\, x_j(t) = b_{j+1} + t\, p_{j+1}$$

and

$$C\,[\, x_j(t) - \sigma_o(t - t_j)s_1\,] + A'_{j+1}\,[\, \hat{v}_j(t) - \sigma_o(t - t_j)\,\hat{s}_2\,] =$$

$$C\, x_j(t) + \sigma_o(t - t_j)A'_j\, s_2 + A'_j\,[\, v_j(t) - \sigma_o(t - t_j)\, s_2\,] =$$

$$C\, x_j(t) + A'_j\, v_j(t) = - (c + t\, q) .$$

By Lemma 4 we have, therefore, $x_{j+1}(t) = x_j(t) - \sigma_o(t - t_j)\, s_1$ which implies

$$h_{1,j+1} = h_{1j} \text{ and } h_{2,j+1} = h_{2j} - \sigma_o\, s_1 . \tag{4.15}$$

Furthermore, $(g_{2j})_1 < 0$ and

$$\omega = (s_2)_1 = s'_1\, A_j\, s_2 = - s'_1\, C\, s_1 < 0 \tag{4.16}$$

imply $\tau_o = (g_{2j})_1 / \omega > 0$.

Now let i be the index of any secondary active constraint. If $a_i' s_1 \geq 0$, then

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i - \sigma_o a_i' s_1$$
$$\leq a_i' h_{2j} - (p)_i \leq 0 .$$

If $a_i' s_1 < 0$, then $i \in I_{4j}$ and

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i - \sigma_o a_i' s_1$$
$$= a_i' s_1 \left(\frac{(p)_i - a_i' h_{2j}}{- a_i' s_1} - \sigma_o \right)$$
$$\leq a_i' s_1 (\tau_o - \sigma_o) < 0 .$$

Therefore, $I_{1,j+1} = \emptyset$.

Case 3:

$(I_{4j} \neq \emptyset , \ \tau_o > 0 , \ \omega = 0)$ or $(I_{4j} \neq \emptyset , \ \tau_o > 0 , \ \omega \neq 0 ,$ and $(g_{2j})_1 / \omega \geq \tau_o)$.

In this case $A_{j+1}' = (a_1, \ldots, a_{\nu-1}, a_k)$. With the same arguments as in the previous section it can be shown that $x_{j+1}(t) = x_j(t) - \tau_o(t - t_j)s_1$, i.e.

$$h_{1,j+1} = h_{1j} \quad \text{and} \quad h_{2,j+1} = h_{2j} - \tau_o s_1 . \tag{4.17}$$

Let i be the index of a secondary active constraint. If $a_i' s_1 \geq 0$, then

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i - \tau_o a_i' s_1$$
$$\leq a_i' h_{2j} - (p)_i \leq 0 .$$

If $a_i' s_1 < 0$, then $i \in I_{4j}$ and

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i - \tau_o a_i' s_1$$
$$= a_i' s_1 \left(\frac{(p)_i - a_i' h_{2j}}{- a_i' s_1} - \tau_o \right) \leq 0 .$$

Hence, $I_{1,j+1} = \emptyset$.

Case 4:

$I_{4j} = \emptyset$ and $\omega = 0$. In this case Step 3.3 of the algorithm applies an

$$A'_{j+1} = (a_1,\ldots,a_{\nu-1},a_k) .$$

Because $\omega = 0$ the equality (4.16) implies $s_1' C s_1 = 0$ which, for the positive semidefinite matrix C , is equivalent to $C s_1 = 0$. Thus

$$C x_j(t_j) = C [x_j(t_j) - \tau_1 s_1] .$$

Furthermore, by the definition of τ_1 in Step 3.3 of the algorithm

$$a_k'(x_j(t) - \tau_1 s_1) = (b)_k + t_j(p)_k .$$

Therefore, it is not difficult to show that

$$x_{j+1}(t) = x_j(t) - \tau_1 s_1 - (t - t_j) \tau_2 s_1$$

$$v_{j+1}(t) = v_j(t) - (t - t_j) \tau_2 s_2$$

is a solution to (4.11). Hence we have

$$h_{1,j+1} = h_{1j} - \tau_1 s_1 , \quad h_{2,j+1} = h_{2j} - \tau_2 s_1 . \tag{4.18}$$

Let i be the index of a secondary active constraint for $x_{j+1}(t_{j+1})$. If this constraint is also a secondary active constraint for $x_j(t_j)$, then $x_{j+1}(t_{j+1}) = x_j(t_j) - \tau_1 s_1$ with $\tau_1 > 0$. Thus $t_{j+1} = t_j$ implies $a_i' s_1 = 0$, i.e.

$$a_i' h_{2,j+1} - (p)_i = a_i' h_{2j} - (p)_i \le 0 .$$

If the i-th constraint is not a secondary active constraint for $x_j(t_j)$, then $i \in I_{6j}$ and

$$\begin{aligned} a_i' h_{2,j+1} - (p)_i &= a_i' h_{2j} - (p)_i - \tau_2 a_i' s_1 \\ &= a_i' s_1 \left(\frac{(p)_i - a_i' h_{2j}}{- a_i' s_1} - \tau_2 \right) \\ &\le 0 . \end{aligned}$$

Therefore, $I_{1,j+1} = \emptyset$.

Lemma 9

For every j_o with $I_{1j_o} = \emptyset$ and $I_{2j_o} \neq \emptyset$ there is $j_1 > j_o$ such that

$$I_{1j} = \emptyset\ ,\quad j=j_o,j_o+1,\ldots,j_1$$

$$I_{2j} \neq \emptyset\ ,\quad j=j_o,j_o+1,\ldots,j_1-1,\quad I_{2j_1} = \emptyset\ .$$

Proof:

If $I_{1j} = \emptyset$ and $I_{2j} \neq \emptyset$ then Step 1.2 and Step 3 of the algorithm are used to determine the matrix A_{j+1} and M_{j+1}. With $t_{j+1} = t_j$ and

$$x_{j+1}(t) = h_{1,j+1} + (t - t_{j+1})\, h_{2,j+1}$$

it follows from formulas (4.18), (4.17), (4.15), and (4.14), respectively, in the proof of Lemma 8 that

$$h_{1,j+1} = h_{1j} - \tau_1 s_1\ ,\quad h_{2,j+1} = h_{2j} - \tau_2 s_1\ ,\quad \tau_1 > 0 \qquad (4.19)$$

or

$$h_{1,j+1} = h_{1j}\ ,\quad h_{2,j+1} = h_{2j} - \sigma s_1\ ,\ \sigma = \min\{\sigma_o, \tau_o\} > 0 \qquad (4.20)$$

or

$$h_{1,j+1} = h_{1j}\ ,\quad h_{2,j+1} = h_{2j}\ . \qquad (4.21)$$

Each vector (h_{1j}, h_{2j}) is uniquely determined by a submatrix A_j of A. Thus there are only finitely many different vectors (h_{1j}, h_{2j}).

First assume that $(h_{1,j+1}, h_{2,j+1})$ is given by (4.19). In this case $s_1' C s_1 = 0$ and, therefore, $C s_1 = 0$. Hence it follows from

$$-s_1'(c + t_j q) = s_1' C x_1(t_j) + s_1' A_j' v_j(t_j) = (v_j(t_j))_1 = (g_{1j})_1 = 0$$

and the equalities

$$C h_{1j} + (t - t_j) C h_{2j} + A_j' q_{1j} + (t - t_j) A_j' g_{2j} = -(c + t_j q) - (t - t_j) q \qquad (4.22)$$

that

$$-q' s_1 = g_{2j}' A_j s_1 = g_{2j}' e_j = (g_{2j})_1 < 0\ .$$

Therefore, we have

$$q' h_{1,j+1} = q' h_{1j} - \tau_1 q' s_1 < q' h_{1j}\ .$$

This implies that $(h_{1,j+1}, h_{2,j+1})$ is determined by (4.19) for at most finitely many j.

Next suppose that $h_{2,j+1}$ is given by formula (4.20). Then $s_1' C s_1 > 0$ and it follows from (4.22) that

$$- q's_1 - h_{2j}' C s_1 = g_{2j}' A_j s_1 = (g_{2j})_1 < 0 .$$

With $Q(h_{2j}) = q'h_{2j} + \frac{1}{2} h_{2j}' C h_{2j}$ we have, therefore,

$$\begin{aligned} Q(h_{2,j+1}) &= Q(h_{2j}) - \sigma(q's_1 + h_{2j}' C s_1) + \frac{\sigma^2}{2} s_1' C s_1 \\ &= Q(h_{2j}) + \sigma(g_{2j})_1 + \frac{\sigma^2}{2} s_1' C s_1 . \end{aligned}$$

Because $(g_{2j})_1 < 0$, we have

$$\sigma(g_{2j})_1 + \frac{\sigma^2}{2} s_1' C s_1 < 0 \quad \text{for} \quad 0 < \sigma \le \frac{(g_{2j})_1}{- s_1' C s_1} .$$

By (4.16), $- s_1' C s_1 = \omega$. Furthermore,

$$\sigma = \min \left\{ \frac{(g_{2j})_1}{\omega} , \tau_o \right\} > 0 .$$

Thus $Q(h_{2,j+1}) < Q(h_{2j})$. In conjunction with the previous results this implies that $h_{2,j+1}$ is determined by (4.20) for at most finitely many j.

In order to complete the proof of the lemma it suffices, therefore, to show that for at most finitely many consecutive indices j we have

$$h_{2,j+1} = h_{2j} . \tag{4.23}$$

It follows immediately from Step 3.1 and Step 4.1 that for every $\hat{j}$ with $h_{2,\hat{j}+1} = h_{2\hat{j}}$, there is a largest integer j^* such that

$$h_{2,j+1} = h_{2j} , \quad j = \hat{j}, \hat{j}+1, \ldots, j^*$$

and

$$a_k \notin \operatorname{span} \left\{ a_i \,\middle|\, i \in J(x_{j^*}) \right\}$$

where k is the index determined in Step 1.1 of the algorithm. Define the set I of integers in such a way that $i \in I$ if and only if

$$a_i' \, x_{\hat{j}}(t_{\hat{j}}) = (b)_i + t_{\hat{j}}(p)_i$$

and

$$a_i \notin \text{span} \left\{ a_i \;\middle|\; i \in J(x_{\hat{j}}) \right\} .$$

Furthermore, define

$$\hat{I} = \left\{ i \;\middle|\; i \in J(x_{\hat{j}}) \, , \; (g_{1\hat{j}})_{\alpha_{\nu j}} = 0 \text{ with } \alpha_{\nu j} = i \right. .$$

Set

$$d = q + C \, h_{2\hat{j}}$$

and consider the minimization problem

$$\min_h \left\{ d'h \;\middle|\; \begin{array}{ll} a_i' \, h = (p)_i \, , & i \in J(x_{\hat{j}}) - \hat{I} \\ a_i' \, h + \lambda_i = (p)_i \, , & \lambda_i \geq 0 \, , \quad i \in \hat{I} \cup I \end{array} \right\} . \tag{4.24}$$

We apply the simplex-method with Bland's [1] minimum index rule to this problem. An initial basic feasible solution is given by

$$h = h_{2\hat{j}} \, , \quad \lambda_i = 0 \, , \quad i \in I .$$

For any $\nu \in \hat{I}$ let $\hat{s}_\nu$ be such that

$$\begin{aligned} a_i' \, \hat{s}_\nu &= 0 \, , \quad i \in J(x_{\hat{j}}) \, , \quad i \neq \nu \\ a_\nu' \, \hat{s}_\nu &= 1 \, . \end{aligned} \tag{4.25}$$

Substituting $h_{2\hat{j}} - \lambda \, \hat{s}_\nu$ into the objective function we observe that λ_ν is a candidate for a basic variable if

$$- d' \hat{s}_\nu = - (q' + h_{2\hat{j}}' \, C) \, s_\nu < 0$$

Replacing j with $\hat{j}$ in (4.22) and using (4.25) we obtain

$$- (q' + h_{2\hat{j}}' \, C) \, \hat{s}_\nu = (\sigma_{2\hat{j}})_\nu .$$

This implies that λ_1 will become a basic variable, where 1 is the same index as the one determined in Step 1.2 of the algorithm. Thus $\hat{s}_1 = s_1$ as defined by Step 3.1 of the algorithm.

A basic variable λ_i is a candidate for becoming a nonbasic variable if $-a_i' s_1 > 0$. If, for all $i \in \hat{I}$,

$$-a_i' s_1 \leq 0 \quad \text{or} \quad \lambda_i > 0$$

then the new basic solution is different from $h_{2\hat{j}}$. If, for some $i \in \hat{I}$,

$$-a_i' s_1 > 0 \quad \text{and} \quad \lambda_i = 0$$

then $i \in I_{4\hat{j}}$ and λ_k will become a nonbasic variable, where k is the index determined in 3.1 of the algorithm

Thus every iteration of the algorithm such that $j \geq j^* + 1$ and $h_{2,j+1} = h_{2j}$ is equivalent with an iteration of the simplex method applied to (4.24). By Bland's rule this implies that there are at most finitely many consecutive j such that (4.23) holds.

Using the above results we can now establish the main theorem.

Theorem

The algorithm determines a finite number of parameter values $t_0, t_1, \ldots, t_\nu$ and vectors h_{1j}, h_{2j}, $j=0,1,\ldots,\nu-1$, such that

i) $\underline{t} = t_0 \leq t_1 \leq \ldots \leq t_\nu$.

ii) For $j=0,1,\ldots,\nu-1$,
$x_j(t) = h_{1j} + (t - t_j)h_{2j}$ is an optimal solution to the problem (2.1) for all t with $t_j \leq t \leq t_{j+1}$.

iii) Either $t_\nu \geq \bar{t}$ or (2.1) has no optimal solution for any $t > t_\nu$.

Proof:

It follows from Lemma 7 through 9 that, for every j, with $t_j = t_{j+1}$, there is some $j_1 > j$ such that

$$t_j = t_{j+1} = \ldots = t_{j_1} < t_{j_1} + 1 . \tag{4.26}$$

Furthermore, for every j,

$$x_i(t) = h_{1j} + (t - t_j)h_{2j}$$

is uniquely determined by the matrix A'_j whose columns are the gradients of the primary active constraints. Since by the definition of t_{j+1} the vector $x_j(t)$ is not an optimal solution to (2.1) for any $t > t_{j+1}$ it follows from (4.26) and the fact that there are only finitely many different submatrices A'_j of A' that the algorithm terminates with some t_ν. If $t < \bar{t}$, termination occurs either with Step 2 or with Step 3 of the algorithm in which case Lemma 6 asserts that the given problem has no optimal solution for any $t > t_\nu$.

References

[1] Bland, G.R. (1977). New finite pivoting rules for the simplex method. Math. Oper. Res. 2, pp. 103-107.

[2] Mangasarian, O. L. (1969). Nonlinear Programming. McGraw-Hill, New York.

[3] Ritter, K. (1962). Ein Verfahren zur Lösung parameter-abhängiger, nichtlinearer Maximum-Probleme. Unternehmensforschung 6, pp. 149-166.

KR/jvs

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER #2197	2. GOVT ACCESSION NO. AD-A100 602	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On Parametric Linear and Quadratic Programming Problems		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Klaus Ritter		8. CONTRACT OR GRANT NUMBER(s) DAAG29-80-C-0041
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 5 - Operations Research
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P.O. Box 12211 Research Triangle Park, North Carolina 27709		12. REPORT DATE March 1981 13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear programming, Quadratic programming, Parametric programming

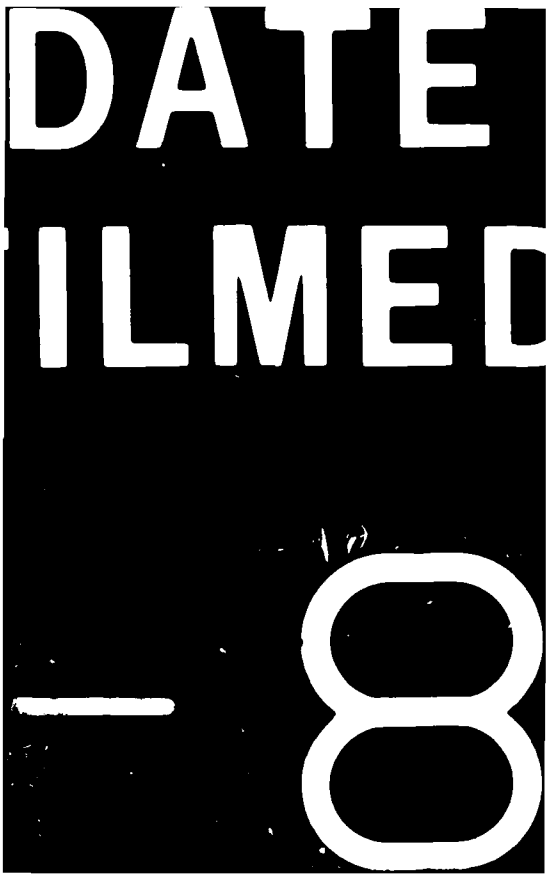
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm is described for determining the optimal solution of parametric linear and quadratic programming problems as an explicit piecewise linear function of the parameter. Each linear function is uniquely determined by an appropriate subset of active constraints. For every critical value of the parameter a new subset has to be determined. A simple rule is given for adding and deleting constraints from this subset.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


DATE
ILMED
8