A PARAMETRIC METHOD FOR SEMI-DEFINITE QUADRATIC PROGRAMS †

M. D. Grigoriadis (*)
K. Ritter (**)

Computer Sciences Technical Report #14

March 1968

(†) Sponsored by the Department of Computer Sciences, University of Wisconsin, under NSF Research Grant GP6070.

(*) and IBM Corporation

(**) and Mathematics Research Center, United States Army, Madison, Wisconsin. Presently at Mathematisches Institut , Technische Hochschule, Karlsruhe, W. Germany.

ABSTRACT

This paper describes a parametric method for solving semi-definite quadratic programs which seems to be well suited for problems with a large number of constraints. All computations are performed by pivotal operations on a tableau, or more efficiently on an inverse, whose size may be considerably smaller when compared to other methods of solution. Updating of this inverse is accomplished by elementary row and column operations. Programming of the proposed algorithm for a computer is facilitated by the ability to make efficient use of the product form of the inverse mechanism of most commercially available linear programming systems. An existing solution to a slightly perturbed problem, if available, may be used as a starting solution for a new problem, with a possible substantial reduction of the required computational effort. Finally, an obvious but rather important advantage of the method is its use in post-optimality studies involving the requirements vector and/or the linear part of the objective function.

1. INTRODUCTION

The Quadratic Programming problem (QP) is defined as:

Maximize

(1.1.1) $$Q(x) = c'x - \frac{1}{2} x' Cx$$

subject to the constraints

(1.1.2) $$Ax \leqq b$$

(1.1.3) $$x \geqq 0$$

where c and x are n-vectors, C is an (n, n) - symmetric positive semi-definite matrix, b is an m-vector and A an (m,n) - matrix.[1)]

In recent years, much attention has been focused on this problem for two main reasons. Although in practice it is used much less frequently than linear programming, quadratic programming has several important applications in business, science and engineering. These include ([18, 4, 6]) problems in portfolio selection, linear regression analysis with inequality constraints on the coefficients, maximization of consumer's utility in the framework of classical consumption theory, profit maximization under resource constraints, quadratic approximation of general convex programs, computational methods of optimal control (minimum energy problems), pattern recognition and others. In addition, being a special case of general convex programs, the quadratic programming problem presents a most desirable feature: the linearity of the objective function gradient. As a

---

1) The assumption of inequality constraints is not restrictive since $Ax = b$ may be represented by $Ax \leqq b$ and the additional constraint $-a'_{m+1} x \leqq -\beta$ where $a'_{m+1} = \sum_{i=1}^{m} a'_i$ and $\beta = \sum_{i=1}^{m} (b)_i$.

result, the Kuhn-Tucker optimality conditions are linear with the additional stipulation that only one of certain pairs of variables may be positive. Algorithms for the solution of quadratic programming problems are abundant in the recent literature on mathematical programming, see e.g. [2, 4, 5, 9, 10, 12, 17, 20]. Some of these methods handle the case of a semi-definite matrix C directly. Others however, insist on a definite C and can handle the semi-definite case only by the perturbation technique first suggested by Barankin and Dorfman [1]. In general, these methods have the use of the Kuhn-Tucker [8] optimality conditions as an auxiliary problem and the application of a modified simplex procedure in common. An initial feasible solution satisfying (1.1.2)-(1.1. is required in most cases. However, if C is definite, instead of an initial feasible solution,[9, 17] require C^{-1} or the equivalent amount of computation. The auxiliary problem is then solved by simplex operations performed on tableaux of size (m+n). Excellent reviews of the above methods may be found in [3, 4, 7] and computational experiments on their relative efficiencies found in [11].

The computational experience with large quadratic programs is, despite all the theoretical development, quite unimpressive when compared to that of linear programming. The methods of Wolfe [18] and Beale [2] have been programmed for large scale computers. Their implementation requires only minor modifications of existing linear programming codes. However, the largest problem solved by these programs, known to the authors, does not exceed 50 variables and 100 constraints.

In [13], Ritter proposes a parametric method for the definite case. This technique amounts to first, obtaining the free maximum x_0 and, if x_0 is not feasible, enlarging the feasible domain so that x_0 becomes a feasible boundary point; and second, considering x_0 as a function of a parameter, reducing this domain to its original definition (1.1.2)-(1.1.3) by successive parametric steps. The method requires explicit definition of the non-negativity restrictions (1.1.3), the inverse C^{-1} and a prohibitive amount of core storage for storing A as well as the partitioned form of an inverse whose size varies between $m+1$ and $m+n$. Furthermore, it cannot utilize the capabilities of an existing linear programming code.

Limited computational experience with this algorithm however, has shown that a parametric approach resembling [13] might be particularly efficient if the obvious deficiencies listed above could be overcome. The parametric method outlined in this paper is a generalization of the basic idea presented in [13] to the semi-definite case. The non-negativity restrictions (1.1.3) are represented and handled implicitly. For problems with $m \leq n$ a constant tableau size of $m+n$, and for problems with $m > n$ a constant tableau size of $2n+1$ is used for all computations. The method is particularly suited for use with the product form of the inverse mechanism of existing LP codes after minor modifications. Updating of the inverse is accomplished by elementary row and column operations. The most important and obvious advantages of this method are its use for post-optimality studies, i.e. varying the requirement vector b and/or the linear part of the objective function, and its ability to utilize an existing solution to a slightly perturbed problem as a starting solution.

In the next section the parametric quadratic programming problem is defined and the basic features of the algorithm, to be detailed in section 3, are summarized. In section 4, the computational aspects, in particular those concerning pivot operations and the product form of the inverse, are discussed. In section 5, the proposed algorithm is extended to post-optimality analysis of quadratic programs. In the final section the validity of the method is demonstrated.

2. THE PROBLEM

We consider a Parametric Quadratic Program (PQP) in which the linear part of the objective function and the right hand side vector are linear functions of a parameter θ, i.e.

Maximize for all θ with $\underline{\theta} \leqq \theta \leqq \overline{\theta}$

(2.1.1) $$Q(x(\theta)) = (c+\theta d)'x - \frac{1}{2}x'Cx$$

subject to:

(2.1.2) $$Ax \leqq b + \theta f$$

(2.1.3) $$x \geqq 0$$

where $\underline{\theta}$, $\overline{\theta}$ define the specified parameter range and d, f are given or predetermined n and m-vectors respectively. Clearly, the above problem is equivalent to QP when $\theta = 0$.

The Kuhn-Tucker necessary optimality conditions [8] for PQP state that if $x = x_0$ is an optimal solution to PQP for a particular $\theta = \theta_0$, then there exist vectors $u = u_0$ and $v = v_0$ such that the following relations are satisfied:

(2.2.1) $$Cx + A'u - v = c + \theta_0 d$$

(2.2.2) $$Ax + y = b + \theta_0 f$$

(2.2.3) $$v'x = 0$$

(2.2.4) $$u'y = 0$$

(2.2.5) $$x \geqq 0$$

(2.2.6) $$y \geqq 0$$

(2.2.7) $$u \geqq 0$$

(2.2.8) $$v \geqq 0$$

where u, v are m and n-vectors of Lagrange multipliers or dual variables, corresponding to (2.1.2) and (2.1.3) respectively, and y is an m-vector of slack variables corresponding to (2.1.2). These conditions are also sufficient in view of a positive semi-definite matrix C . The relations (2.2.1) - (2.2.8) will be referred to as the "Auxiliary Problem" (AP).

We note that AP is linear except for (2.2.3) - (2.2.4) which are usually referred to as the "complementary slackness" conditions. Assuming that a solution to AP is known for some $\theta = \theta_0$, our aim is to obtain subsequent solutions for $\theta < \theta_0$, provided they exist, until θ is reduced to zero. Thus, the corresponding solution will be optimal for QP .

An important aspect of the proposed algorithm is its ability to determine off-normal conditions in QP or PQP from information available in the simplex tableau of AP . Suppose for example QP or PQP for some $\theta = \theta_k$, have no feasible solution. This fact may be determined from the corresponding AP by a simple test. Similarly, if QP or PQP for some $\theta = \theta_k$ have no optimal solution, or have an unbounded solution (provided that they have non-empty feasible domains), this fact may be detected from the corresponding AP .

3. THE ALGORITHM

Our first step toward obtaining an optimal solution to a given QP is to choose an x_0 and determine d, f and θ so that x_0 is an optimal solution to PQP . We propose to begin with $x_0 = 0$. If we let

$$(3.1.1) \qquad (d)_i = \begin{cases} -1 & \text{if } (c)_i \geqq 0 \\ 0 & \text{otherwise} \end{cases} ; \quad i = 1, \ldots, n$$

$$(3.1.2) \qquad (f)_i = \begin{cases} 1 & \text{if } (b)_i \leqq 0 \\ 0 & \text{otherwise} \end{cases} ; \quad j = 1, \ldots, m$$

and choose

$$(3.1.3) \qquad \theta_0 = \max \; \{0, (c)_i \, , \; - (b)_j\}; \; i = 1, \ldots, n; \; j = 1, \ldots, m,$$

we have vectors d and f such that for $\theta = \theta_0$, the point $x_0 = 0$ is an optimal solution to PQP. Furthermore, if $\theta_0 = 0$, then $x_0 = 0$ solves QP as well as PQP . Assuming $\theta_0 > 0$, we seek to obtain a solution to QP by parametrically solving PQP for values of $\theta < \theta_0$ until a solution, if it exists, is obtained for $\theta = 0$. Such successive solutions are obtained using simplex tableaux of the corresponding AP. The side conditions (2.2.3) - (2.2.4) are handled by the proper choice of pivot. The pivot operation is regarded as an exchange of an active constraint and an inactive one, or as it is frequently portrayed, as an exchange of a basic slack variable (corresponding to the inactive constraint) and a non-basic one (corresponding to the active constraint). Thus, for $\theta = \theta_0$ equations (2.2.2) give

$y_0 = b + \theta f$; and $y_0 \geqq 0$ due to the construction of the vector f. Similarly, (2.2.1) implies that $v_0 = -(c + \theta_0 d)$. Considering the construction of the vector d by (3.1.1), we have $v_0 \geqq 0$. Therefore, $x_0 = 0$, $u_0 = 0$, y_0 and v_0 is a solution of AP for $\theta = \theta_0$. In the terminology of the simplex method we regard v_0 and y_0 as "basic" in the tableau of AP.

Before outlining the details of the proposed algorithm, we examine the structure of AP and the composition of its basis at any parametric step corresponding to $\theta \leqq \theta_0$. We first note that since PQP has n variables, at most n constraints of the form (2.1.2) are needed to determine the optimal solution of PQP for any particular θ. In terms of the definition of AP this implies that at any time at most n components of u can be basic. Hence, we can assume that in each solution of AP at least $(m-n)$ components of y, corresponding to the inactive constraints, are basic. In degenerate cases, some of these may be at zero level.

Based on the above observation, let us partition the rows of A in the following fashion. Assume that for a particular $\theta = \theta_k \leqq \theta_0$ we have $p\ (\leqq n)$ active constraints of the form (2.1.2). Denote by A^c the $(n+1, n)$ - submatrix of A which includes the p active rows and $(n-p+1)$ of the inactive ones. These will be referred to as the "current" rows of A. Similarly, let A^s denote the $(m-n-1, n)$ - submatrix of A, consisting of the remaining inactive constraint rows of A. These will be referred to as the "stand by" rows of A. The corresponding partitioning of the vectors u and y gives (u^c, u^s) and (y^c, y^s) respectively. This is shown in Figure 1.

$$
\begin{array}{r l}
 & \begin{array}{cccc|cc} x, & u^c, & v, & y^c & u^s, & y^s \end{array} \\
\begin{array}{r} n \{ \\ n+1 \{ \\ m-n-1 \{ \end{array} &
\left[\begin{array}{cccc|cc}
C & A^{c\prime} & -I & 0 & A^{s\prime} & 0 \\
A^c & 0 & 0 & I & 0 & 0 \\
\hline
A^s & 0 & 0 & 0 & 0 & I
\end{array}\right]
\begin{array}{l} = c + \theta d \\ = b^c + \theta f^c \\ = b^s + \theta f^s \end{array} \\
 & \begin{array}{cccccc} \underbrace{}_{n} & \underbrace{}_{n+1} & \underbrace{}_{n} & \underbrace{}_{n+1} & \underbrace{}_{m-n-1} & \underbrace{}_{m-n-1} \end{array}
\end{array}
$$

Figure 1 - Matrix structure of Auxiliary Problem (AP)

We observe that since the rows of A^s are all inactive, y^s must be basic and $u^s = 0$. Therefore, all information on the solution of AP may essentially be obtained from the (2n+1, 4n+1) - submatrix enclosed within dotted lines in Figure 1. Since, at any time, A^s contains only inactive rows, we will perform the necessary variable exchange (pivot) operations on the "Reduced Auxiliary Problem" (RAP) defined as:

(3.2.1) $$Cx + A^{c\prime} u^c - v = c + \theta d$$

(3.2.2) $$A^c x + y^c = b^c + \theta f^c$$

(3.2.3) $$x, u^c, v, y^c \geqq 0$$

always taking into account the side conditions:

(3.2.4) $$v' x = 0$$

(3.2.5) $$u^{c\prime} y^c = 0 \qquad (\text{and } u^{s\prime} y^s = 0) .$$

If, during the course of the algorithm a constraint in A^s becomes active, then the current partitioning of A will be updated to include this constraint in A^c in exchange for a currently inactive one in A^c, which is brought into A^s. This operation, also handled by pivoting, requires an effort almost equivalent to an ordinary pivot step. The structure of the working basis at any parametric step corresponding to $\theta = \theta_k \leqq \theta_0$ is given by:

$$(3.3) \qquad B_k = \begin{array}{c} x_{1k}, \quad v_{1k}, \quad u^c_{1k}, \quad y^c_{1k} \\ \left[\begin{array}{cccc} C_{11} & 0 & A^{c\prime}_{11} & 0 \\ C_{21} & -I & A^{c\prime}_{12} & 0 \\ A^c_{11} & 0 & 0 & 0 \\ A^c_{21} & 0 & 0 & I \end{array}\right] \end{array} \begin{array}{l} \} \ p \\ \} \ n-p \\ \} \ q \\ \} \ n+1-q \end{array} \begin{array}{l} \left.\right\} n \\ \left.\right\} n+1 \end{array} \left.\right\} 2n+1$$

where x_{1k}, v_{1k}, u^c_{1k}, y^c_{1k} denote the basic variables corresponding to the basis B_k of RAP and where the following partitions have been used:

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad ; \qquad A^c = \begin{pmatrix} A^c_{11} & A^c_{12} \\ A^c_{21} & A^c_{22} \end{pmatrix}$$

$$x' = (x_1', x_2'); \quad v' = (v_1', v_2'); \quad u^{c\prime} = (u^{c\prime}_1, u^{c\prime}_2); \quad y^{c\prime} = (y^{c\prime}_1, y^{c\prime}_2)$$

Corresponding partitions of the right hand side are:

$$(c + \theta d)' = (c_1' + \theta d_1', \ c_2' + \theta d_2') \ ; \quad (b^c + \theta f^c) = (b^{c\prime}_1 + \theta f^{c\prime}_1, \ b^{c\prime}_2 + \theta f^{c\prime}_2)$$

and of the inactive "stand-by" constraints:

$$A^s = (A^s_1, \ A^s_2)$$

In addition to the above, we define the following index sets for later reference. Let $J = \{1, \ldots, n\}$ be the index set corresponding to the variables x, $I = \{1, \ldots, m\}$ be the index set corresponding to the rows of A, and $I^c \subseteq I$, $I^s \subset I$ with $I^c \cup I^s = I$ be the index sets corresponding to the rows of A^c and A^s respectively. Furthermore let $J_B \subseteq J$ contain all indices of the components of the p-vector x_1, $I_B^c \subseteq I^c$ contain those indices corresponding to the active constraint rows or to the q-vector u_1^c. Similarly, let $\bar{J}_B = J - J_B$ be the indices corresponding to the (n-p)-vector v_1 and $\bar{I}_B^c = I^c - I_B^c$ be those corresponding to the $(n + 1 - q)$-vector y_1^c.

Using the above notation we may refer to the basic variables (with respect to a basis B_k; $\theta = \theta_k$) as $(x_{1k})_j$ for $j \in J_{B_k}$; $(v_{1k})_j$ for $j \in \bar{J}_{B_k}$; $(u_{1k}^c)_j$ for $j \in I_{B_k}^c$ and $(y_{1k}^c)_j$ for $j \in \bar{I}_{B_k}^c$. We consider $(y_k^s)_j$ for $j \in I_{B_k}^s$ as "implicitly basic" since they do not appear in the RAP basis but are basic in the tableau of AP.

Now assume that for $\theta = \theta_k$ an optimal solution $x_k(\theta_k)$ has already been obtained. We wish to determine the smallest θ, say $\theta = \theta_\ell \leqq \theta_k$, such that $x_k(\theta)$, $\theta_\ell \leqq \theta \leqq \theta_k$, remains both feasible and optimal. Considering the (2n+1)-order RAP basis B_k and its inverse B_k^{-1}, we can express the current solution as:

$$(3.4.1) \qquad \begin{pmatrix} x_{1k}(\theta) \\ v_{1k}(\theta) \\ u_{1k}^c(\theta) \\ y_{1k}^c(\theta) \end{pmatrix} = B_k^{-1} \begin{pmatrix} c_1 + \theta d_1 \\ c_2 + \theta d_2 \\ b_1^c + \theta f_1^c \\ b_2^c + \theta f_2^c \end{pmatrix} \qquad ; \ \theta = \theta_k$$

For convenience we write

(3.4.2) $$x_{1k}(\theta) = p_1 + \theta p_2$$

(3.4.3) $$v_{1k}(\theta) = p_3 + \theta p_4$$

(3.4.4) $$u^c_{1k}(\theta) = p_5 + \theta p_6$$

(3.4.5) $$y^c_{1k}(\theta) = p_7 + \theta p_8$$

where the composition of the p_j can be readily identified from (3.4.1). The implicitly basic slack variables $y^s_k(\theta)$ corresponding to the "stand-by" constraints A^s may be expressed as

$$y^s_k(\theta) = b^s + \theta f^s - A^s_1 \, x_{1k}(\theta)$$

or

$$y^s_k(\theta) = (b^s - A^s_1 \, p_1) + \theta(f^s - A^s_1 \, p_2)$$

where we write for convenience:

(3.4.6) $$y^s_k(\theta) = p_9 + \theta p_{10}$$

At this point two questions must be answered. The first is that of optimality: What is the smallest value of the parameter, say $\theta = \tilde{\theta}_1 \leqq \theta_k$, for which the optimality conditions

(3.5.1) $$v_{1k}(\theta) \geqq 0$$

(3.5.2) $$u^c_{1k}(\theta) \geqq 0$$

are satisfied? For (3.5.1), using (3.4.3), we have:

(3.6.1) $$\tilde{\theta}'_{\ell} = -\frac{(p_3)_{\mu'}}{(p_4)_{\mu'}} = \max\left\{ -\frac{(p_3)_j}{(p_4)_j} \;\middle|\; (p_4)_j > 0 \text{ and } j \in I^c_{B_k} \right\}$$

Similarly, for (3.5.2), using (3.4.4), we obtain:

(3.6.2) $$\tilde{\theta}''_{\ell} = -\frac{(p_5)_{\mu''}}{(p_6)_{\mu''}} = \max\left\{ -\frac{(p_5)_j}{(p_6)_j} \;\middle|\; (p_6)_j > 0 \text{ and } j \in \bar{J}_{B_k} \right\}$$

Thus, for optimality, the smallest value of θ is:

$$\tilde{\theta}_{\ell} = \max \{\tilde{\theta}'_{\ell}, \tilde{\theta}''_{\ell}\}$$

and

$$\mu = \begin{cases} \mu' & \text{if } \tilde{\theta}_{\ell} = \tilde{\theta}'_{\ell} \\ \mu'' & \text{if } \tilde{\theta}_{\ell} = \tilde{\theta}''_{\ell} \end{cases}$$

The second question consists of determining the smallest value of the parameter, say $\theta = \hat{\theta}_{\ell} \leqq \theta_k$ such that the feasibility conditions:

(3.7.1) $$x_{1k}(\theta) \geqq 0$$

(3.7.2) $$y^c_{1k}(\theta) \geqq 0$$

(3.7.3) $$y^s_k(\theta) \geqq 0$$

are satisfied. For (3.7.1), using (3.4.2), we have

(3.8.1) $$\hat{\theta}'_{\ell} = -\frac{(p_1)_{\rho'}}{(p_2)_{\rho'}} = \max\left\{ -\frac{(p_1)_j}{(p_2)_j} \;\middle|\; (p_2)_j > 0 \text{ and } j \in J_{B_k} \right\}$$

Similarly, using (3.4.5)-(3.4.6) with (3.7.2)-(3.7.3) respectively:

(3.8.2) $$\hat{\theta}''_{\ell} = -\frac{(p_7)_{\rho''}}{(p_8)_{\rho''}} = \max\left\{ -\frac{(p_7)_j}{(p_8)_j} \;\middle|\; (p_8)_j > 0 \text{ and } j \in \bar{I}^c_{B_k} \right\}$$

and

$$(3.8.3) \qquad \hat{\theta}_\ell''' = -\frac{(p_9)_{\rho'''}}{(p_{10})_{\rho'''}} = \max\left\{ -\frac{(p_9)_j}{(p_{10})_j} \;\middle|\; (p_{10})_j > 0 \text{ and } j \in I^s_{B_k} \right\}$$

Therefore, for feasibility the smallest value of θ is:

$$\hat{\theta}_\ell = \max \; \{\hat{\theta}_\ell', \hat{\theta}_\ell'', \hat{\theta}_\ell''\}$$

and

$$\rho = \begin{cases} \rho' & \text{if } \hat{\theta}_\ell = \hat{\theta}_\ell' \\ \rho'' & \text{if } \hat{\theta}_\ell = \hat{\theta}_\ell'' \\ \rho''' & \text{if } \hat{\theta}_\ell = \hat{\theta}_\ell''' \end{cases}$$

Finally the smallest θ for which the present solution remains both optimal and feasible is:

$$(3.9) \qquad \theta_\ell = \max\{\tilde{\theta}_\ell, \hat{\theta}_\ell\} = \max\{\tilde{\theta}_\ell', \tilde{\theta}_\ell'', \hat{\theta}_\ell', \hat{\theta}_\ell'', \hat{\theta}_\ell'''\} .$$

The next step is to investigate the nature of the AP basic solution for $\theta = \theta_\ell - \varepsilon > 0$ for a small $\varepsilon > 0$. This is the crucial part in performing a parametric step. Depending on which one of the five limiting values of θ defines θ_ℓ in (3.9), an appropriate basis change must be performed in order to restore optimality or feasibility. Such a basis change has the usual geometric interpretation. It entails updating the status of a constraint, be it a non-negativity restriction or an ordinary constraint, from "active" to "inactive" and vice-versa. As in parametric programming for linear programs, such exchanges for a particular limiting value of θ, cause no change in the current value of the objective function since both the entering and exiting variables are at zero level. In linear programming, there is considerable flexibility in the choice of the entering variable. In our case, however, once the exiting variable is defined by (3.6) or (3.8), the entering variable is uniquely determined by (2.2.3)-(2.2.4). Such exchanges are performed by pivot operations.

We now examine the types of pivot operations which will restore feasibility (Case I) or optimality (Case II) for $\theta = \theta_\ell - \varepsilon$. Characteristically, in Case I we would like to reclassify a currently (i.e. for $\theta = \theta_\ell$) "active" constraint as "inactive", and in Case II a currently "inactive" constraint as "active". Thus, one of the following operations would have to be performed for $\theta = \theta_\ell$:

Case I

$(\theta_\ell = \tilde{\theta}_\ell)$. The constraint to become inactive is a:

(i) non-negativity restriction $(\theta_\ell = \tilde{\theta}_\ell = \tilde{\theta}'_\ell \leqq \theta_k)$

By (3.6.2), we must have $(v_{1k})_\mu = 0$ for at least one $\mu \in \bar{J}^c_{B_k}$.

In view of (2.2.3), we seek to replace $(v_{1k})_\mu$ by its complementary variable $(x_{2k})_\mu$.

(ii) ordinary constraint $(\theta_\ell = \tilde{\theta}_\ell = \tilde{\theta}''_\ell \leqq \theta_k)$

From (3.6.1), we must have $(u^c_{1k})_\mu = 0$ for at least one $\mu \in I^c_{B_k}$.

In view of (2.2.4) we seek to replace $(u^c_{1k})_\mu$ by its complementary variable $(y^c_{2k})_\mu$.

<u>Case II</u>

$(\theta_\ell = \hat{\theta}_\ell)$. The constraint to become active is a:

(i) <u>non-negativity restriction</u> $(\theta_\ell = \hat{\theta}_\ell = \hat{\theta}'_\ell \leqq \theta_k)$

From (3.8.1), we must have $(x_{1k})_\rho = 0$ for at least one $\rho \in J_{B_k}$. In view of (2.2.3) we seek to replace $(x_{1k})_\rho$ by its complementary variable $(v_{2k})_\rho$.

(ii) <u>ordinary constraint in A^C</u> $(\theta_\ell = \hat{\theta}_\ell = \hat{\theta}''_\ell \leqq \theta_k)$

From (3.8.2), we must have $(y^C_{1k})_\rho = 0$ for at least one $\rho \in I^C_{B_k}$. In view of (2.2.4) we seek to replace $(y^C_{1k})_\rho$ by its complementary variable $(u^C_{2k})_\rho$.

(iii) <u>ordinary constraint in A^S</u> $(\theta_\ell = \hat{\theta}_\ell = \hat{\theta}'''_\ell \leqq \theta_k)$

From (3.8.3), we must have $(y^S_k)_\rho = 0$ for at least one $\rho \in I^S_{B_k}$. We recall that although y^S is not in the basis of RAP, it is always considered as basic in AP since A^S is composed of only inactive constraints. However, now that the ρth constraint is A^S will become active, we must perform the following operations:

a) Update the definition of the "current" and "stand-by" constraints, i.e. update the sets I^C and I^S, by defining the ρth constraint presently in A^S as "current", in exchange for an inactive constraint presently in A^C, say the τth. Such a constraint will always be present in A^C since not more than n of the (n+1) constraints in A^C

may be active. The updated partitioning of A into A^c and A^s defines a new RAP. It is obtained from the current RAP by replacing the elements of the row corresponding to the τth constraint by the representation of the ρth constraint (which is to be made active) and furthermore, by replacing the basic variable $(y^c_{1k})_\tau$ by the implicitly basic variable $(y^s_k)_\rho$. One should note that in the new RAP, the updated partitioning of y into y^c and y^s requires that $(y^s_k)_\rho$ be denoted by $(y^c_{1k})_\tau$. This notational liberty should not cause confusion. In the new RAP, $(y^c_{1k})_\tau$ is at zero level and must be removed from the basis by (b) below.

b) We seek to replace the currently basic $(y^c_{1k})_\tau$ by its complementary variable $(u^c_{1k})_\rho$. This step is the same as (ii) above.

Any of the exchanges described above may be easily performed by a "pivot operation". Lemma 2 in section 6 guarantees the non-positivity of the pivot element. Thus, if the pivot element is negative, we perform one pivot step so that the sought exchange of variables is accomplished. We then return to (3.4.1). The mechanics of pivoting or, equivalently, updating the current inverse of the RAP basis, are outlined in section 4 . On the other hand, if the pivot element is zero, the sought exchange of variables is more complex and requires a pair of pivot steps. The existence of a pair of pivots is demonstrated in Theorem 2. In the following paragraphs the question of a zero pivot (in Cases I and II) and its remedy are discussed in detail.

Case I:

If the pivot element is zero, it can be shown (Lemma 3) that PQP has an infinite number of optimal solutions for $\theta = \theta_\ell$. However, by means of simple examples it may be shown that not all elements $\bar{x}_k$ of this infinite set of optimal solutions need have the property that there exists a function $x(\theta)$ such that $\bar{x}_k = x_k(\theta_\ell)$ and $x(\theta)$ are optimal solutions to PQP for some interval $\theta_p \leqq \theta \leqq \theta_\ell$ with $\theta_p < \theta_\ell$. The existence of a zero pivot indicates that $\bar{x}_k$ does not have this property. In order to continue with our parametric procedure we have to determine an optimal solution $\hat{x}_k$ (if it exists) for which there exists a function $x(\theta)$ with the previous property.

This is referred to, in this paper, as the "Search Procedure" which is perhaps a misnomer since the term usually implies a more complicated sequence of computations such as a one dimensional optimization procedure. Our search procedure is defined as follows:

For Case I - (i), where a non-negativity restriction is to become inactive, i.e. $(v_{1k})_\mu = 0$, we let a' denote the μ th row of the matrix $(C'_{12}, C'_{22}, A^{c'}_{12}, A^{c'}_{22})$. Furthermore, we let s^1_k and s^2_k denote the first p and the last (n + 1 - q) components of the vector $-B^{-1}_k a$, respectively.

For Case I - (ii), where an ordinary constraint is to become inactive, i.e. $(u^c_{1k})_\mu = 0$, we let e_μ denote an (n + 1)-vector which has 1 as its μ th component and zero elsewhere. Let $\hat{s}^1_k$ and $\hat{s}^2_k$ denote the first p and the last (n + 1 - q) components of the vector $B^{-1}_k e_\mu$, respectively.

Now we define the directions:

$$\bar{s}_k = \begin{cases} (s^1_k,\ s) & \text{if Case I - (i) applies} \\ (\hat{s}^1_k,\ 0) & \text{if Case I - (ii) applies} \end{cases} \tag{3.10.1}$$

where s is an (n - p)-vector with -1 as its μ th component and zero elsewhere;

and

$$s_k = \begin{cases} s^2_k & \text{if Case I - (i) applies} \\ \hat{s}^2_k & \text{if Case I - (ii) applies} \end{cases} \tag{3.10.2}$$

By Lemma 3, $x(\lambda) = \bar{x}_k + \lambda \bar{s}_k$ is optimal for all λ for which it remains feasible. We wish to determine the smallest λ for which $x(\lambda)$ is feasible.

Since by construction of $\bar{s}_k$, all the constraints active at $\bar{x}_k$ are satisfied by $x(\lambda)$, $\lambda \leqq 0$, we have only to consider the following three cases:

a) the inactive constraints in A^c must be satisfied, i.e. we must have

$$y^c_{1k}(\theta_\ell) + \lambda s_k \geqq 0$$

which gives

$$\text{(3.11.1)} \qquad \lambda_1 = \max_j \left\{ - \frac{(y^c_{1k})_j}{(s_k)_j} \;\middle|\; (s_k)_j > 0 \text{ and } j \in \bar{I}^c_{B_k} \right\}$$

the maximum occurring for some $j = \sigma_1$. If $s_k \leqq 0$ for all $j \in \bar{I}^c_{B_k}$, then $\lambda_1 = -\infty$

b) the inactive non-negativity restrictions must be satisfied, i.e. we must have:

$$(\bar{x}_k + \lambda \bar{s}_k)_j \geqq 0 \text{ for all } j \in J_{B_k}$$

or

$$\text{(3.11.2)} \qquad \lambda_2 = \max_j \left\{ - \frac{(\bar{x}_k)_j}{(\bar{s}_k)_j} \;\middle|\; (\bar{s}_k)_j > 0 \text{ and } j \in J_{B_k} \right\}$$

the maximum occurring for some $j = \sigma_2$. If $(\bar{s}_k)_j \leqq 0$ for all $j \in J_{B_k}$, then $\lambda_2 = -\infty$.

c) the (inactive) constraints A^s must be satisfied, i.e. we must have:

$$A^s(\bar{x}_k + \lambda \bar{s}_k) \leqq b^s + \theta_\ell f^s$$

or

$$\lambda_3 = \max_j \left\{ \frac{(y_k^s)_j}{(A^s\bar{s}_k)_j} \;\middle|\; (A^s\bar{s}_k)_j < 0 \text{ and } j \in I_{B_k}^s \right\}$$

the maximum occurring for some $j = \sigma_3$. If $(A^s\bar{s}_k)_j \geqq 0$ for all $j \in I_{B_k}^s$, then $\lambda_2 = -\infty$

We let

(3.12.1) $\underline{\lambda} = \max \{\lambda_1, \lambda_2, \lambda_3\}$

(3.12.2) $\sigma = \{\sigma_j \mid \lambda_j = \underline{\lambda}\}$

If $\underline{\lambda} = -\infty$, then there exists no optimal solution to QP (Theorem 1). If a feasible solution to QP exists, then for $\theta < \theta_\ell \geqq 0$, PQP (and QP) has an unbounded solution. (Remark 2 in section 6).

If $-\infty < \underline{\lambda} \leqq 0$, then $\hat{x}_k = x(\underline{\lambda}) = \bar{x}_k + \underline{\lambda}\,\bar{s}_k$ is an optimal solution to PQP for $\theta = \theta_\ell$, which can be used to continue the parametric procedure. The transition from $\bar{x}_k$ to $\hat{x}_k$ causes the σth constraint to become active instead of the μth one. The complete solution of RAP, corresponding to the optimal vector $\hat{x}_k$, can be obtained from the old one by means of a pair of pivot operations (Theorem 2).

In terms of variable exchanges, we may say that in Case I-(i) we originally sought to exchange $(v_{1k})_\mu$ by $(x_{1k})_\mu$ and found that this was not possible, due to a zero pivot element. Now we can exchange

(3.13.1) $(v_{1k})_\mu$ by $(u_{2k}^c)_\sigma$ <u>and</u> $(y_{1k}^c)_\sigma$ by $(x_{2k})_\mu$ if $\sigma = \sigma_1$

which if replaced by the ρth constraint, insures that the so altered set of active constraints is linearly independent and RAP has a solution $x_{k+1}(\theta)$ for some interval $\theta_q \leqq \theta \leqq \theta_\ell$ with $\theta_q < \theta_\ell$ and $x_{k+1}(\theta_\ell) = x_k(\theta_\ell)$. This procedure, which will be referred to as the "Constraint Replacement Procedure", is outlined below.

Let the ρth constraint be represented by:

$$a'x = a_1'x_1 + a_2'x_2 \leqq \beta \tag{3.15.1}$$

Since, by Lemma 2, the above constraint is linearly dependent on the active constraints at $\bar{x}_k$, there exists an (n-p)-vector z_1 and a q-vector z_2 such that

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 & A_{11}^{c\prime} \\ -I & A_{12}^{c\prime} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \tag{3.16.1}$$

Using the current partitioning (3.3) of B_k, it may be easily verified that $(0, z_1', z_2', 0)$ is the unique solution of the system $z'B_k = (a_1', a_2', 0, 0)$. Hence, B_k^{-1} being known, (z_1', z_2') can be determined conveniently from

$$(0, z_1', z_2', 0) = (a_1', a_2', 0, 0)\ B_k^{-1} \tag{3.16.2}$$

With (z_1', z_2') known, Lemma 4 provides us with a simple test to identify the μth constraint to be made inactive or to detect that PQP has no feasible solution for $\theta < \theta_\ell$. Thus, if for $z' = (0, z_1', z_2', 0)$,

$$(z)_j \leqq 0 \text{ for all } j \in I_{B_k}^c, \tag{3.17.1}$$

we conclude that no feasible solution to QP exists. On the other hand, if z has at least one positive component, we obtain

(3.18.1) $$\nu_0^1 = \frac{(v_{1k})_{\mu_1}}{(z_1)_{\mu_1}} = \min_j \left\{ \frac{(v_{1k})_j}{(z_1)_j} \;\middle|\; (z_1)_j > 0, \text{ and } j \in \overline{J}_{B_k} \right\}$$

(3.18.2) $$\nu_0^2 = \frac{(u^c_{1k})_{\mu_2}}{(z_1)_{\mu_2}} = \min_j \left\{ \frac{(u^c_{1k})_j}{(z_2)_j} \;\middle|\; (z_2)_j > 0, \text{ and } j \in I^c_{B_k} \right\}$$

and

(3.18.3) $$\nu_0 = \min \{\nu_0^1, \nu_0^2\}$$
$$\mu = \begin{cases} \mu_1 & \text{if } \nu_0 = \nu_0^1 \\ \mu_2 & \text{if } \nu_0 = \nu_0^2 \end{cases}$$

which identify the μth constraint to be made inactive.

We consider two cases, depending on whether (3.15.1) is a "non-negativity restriction" or an "ordinary constraint".

In Case II - (i) the ρth non-negativity restriction, which is represented by letting $a_1 = 0$, $a_2 = (e)_\rho$ in (3.15.1) and (3.16.2), is to become inactive. However, the exchange of $(x_{1k})_\rho$ by $(v_{2k})_\rho$ cannot be carried out due to a zero pivot. Provided (3.17.1) is not satisfied, one of the following pair of pivots, performed in the indicated order, is always possible (Theorem 2):

(3.19.1) $(x_{1k})_\rho$ by $(v_{2k})_\mu$ <u>and</u> $(x_{1k})_\mu$ by $(v_{2k})_\rho$ if $\mu = \mu_1$

(3.19.2) $(x_{1k})_\rho$ by $(y^c_{2k})_\mu$ <u>and</u> $(u^c_{1k})_\mu$ by $(v_{2k})_\rho$ if $\mu = \mu_2$

In Case II - (ii), the ρth ordinary constraint, which is represented by letting $a_1 = a_{1\rho}$ and $a_2 = a_{2\rho}$ in (3.15.1) and (3.16.2), is to become inactive.

However, the exchange of $(u^c_{1k})_\rho$ by $(y^c_{2k})_\rho$ cannot be accomplished due to a zero pivot. Provided (3.17.1) is not satisfied, one of the following pair of pivots is possible (Theorem 2)

(3.20.1) $(u^c_{1k})_\rho$ by $(v_{2k})_\mu$ <u>and</u> $(x_{1k})_\mu$ by $(y^c_{2k})_\rho$ if $\mu = \mu_1$

(3.20.2) $(u^c_{1k})_\rho$ by $(y^c_{2k})_\mu$ <u>and</u> $(u^c_{1k})_\mu$ by $(y^c_{2k})_\rho$ if $\mu = \mu_2$

Thus the appropriate pivot or pair of pivot steps, specified by Case I or II above, is executed by updating the current inverse using the procedure outlined in section 4 . The updated basis inverse is then used to continue the algorithm from (3.4.1).

A concise summary of the parametric algorithm outlined in the previous paragraphs is as follows:

<u>Step 0.</u> Use (3.1) to construct vectors d and f such that for $\theta = \theta_0$, the point $x_0(\theta_0) = 0$ is an optimal solution to PQP. Construct the basis B_0 by (3.3), define the sets J_{B_0}, $\bar{J}_{B_0}$, $I^c_{B_0}$, $\bar{I}^c_{B_0}$, $I^s_{B_0}$ and obtain B_0^{-1}. Let $\theta_k = \theta_0$, $B_k = B_0$ and go to step 1 .

<u>Step 1.</u> If $\theta_k = 0$, then B_k is optimal and $x_k(0)$ is an optimal solution to QP. <u>Terminate.</u> If $\theta_k > 0$, compute vectors p_j; $j = 1, \ldots, 10$ using (3.4). Apply (3.6), (3.8) and (3.9) to obtain $\theta_\rho \leqq \theta_k$. Go to step 2. (See Remark 3.1)

Step 2. 2.1. (Case I(i)) If $\theta_\ell = \tilde{\theta}'_\ell$, try to exchange $(v_{1k})_\mu$ by $(x_{2k})_\mu$. If the pivot is negative, perform one pivot step to obtain B_ℓ^{-1}, go to step 5. If the pivot is zero go to step 3. Upon return (from step 3) perform one of the pairs of pivots (3.13.1), (3.13.2), or after exchanging a constraint, perform the pair of pivots (3.13.3) to obtain B_ℓ^{-1}. Go to step 5 .

2.2. (Case I(ii)) If $\theta_\ell = \tilde{\theta}''_\ell$, try to exchange $(u^c_{1k})_\mu$ by $(y^c_{2k})_\mu$. If the pivot is negative, perform one pivot step to obtain B_ℓ^{-1}, go to step 5. If the pivot is zero, go to step 3. Upon return (from step 3) perform one of the pairs of pivots (3.14.1), (3.14.2), or after exchanging a constraint, perform the pair of pivots (3.14.3), to obtain B_ℓ^{-1}. Go to step 5.

2.3. (Case II(i)) If $\theta_\ell = \hat{\theta}'_\ell$, try to exchange $(x_{1k})_\rho$ by $(v_{2k})_\rho$. If the pivot is negative, perform one pivot step to obtain B_ℓ^{-1}, go to step 5. If the pivot is zero, go to step 4. Upon return (from step 4) perform one of the pairs of pivots (3.19.1) or (3.19.2) to obtain B_ℓ^{-1}. Go to step 5.

2.4. (Case II(ii)) If $\theta_\ell = \hat{\theta}''_\ell$, try to exchange $(y^c_{1k})_\rho$ by $(u^c_{2k})_\rho$. If the pivot is negative, perform one pivot step to obtain B_ℓ^{-1}, go to step 5. If the pivot is zero, go to step 4. Upon return (from step 4) perform one of the pairs of pivots (3.20.1) or (3.20.2) to obtain B_ℓ^{-1}. Go to step 5.

2.5. (Case II(iii)) If $\theta_\ell = \hat{\theta}_\ell'''$, exchange the ρ^{th} constraint in A^s by the τ^{th} constraint in A^c. This is accomplished by a special pivot step (see 4.1.4) and subsequent updating of B_k^{-1}. Then, try to exchange $(y_{1k}^c)_\tau$ by $(u_{2k}^c)_\rho$. If the pivot is negative, perform one pivot step to obtain B_ℓ^{-1}, go to step 5. If the pivot is zero, go to step 4. Upon return (from step 4) perform one of the pairs of pivots (3.20.1) or (3.20.2) to obtain B_ℓ^{-1}. Go to step 5.

Step 3. ("Search procedure"). If entering from step (2.1), let $\bar{s}_k = -B_k^{-1}(e)_\mu$. If entering from step (2.2), let $\bar{s}_k = -B_k^{-1}(e)_{\mu+n}$. Compute $\lambda_1, \lambda_2, \lambda_3$ by (3.11) and $\underline{\lambda}$, σ by (3.12). If $\underline{\lambda} = -\infty$, no optimal solution to QP exists. (If QP has a non-empty feasible domain, then $\underline{\lambda} = -\infty$ implies that QP has an unbounded solution). Terminate. If $-\infty < \underline{\lambda} \leqq 0$, return.

Step 4. ("Constraint replacement procedure"). If entering from step (2.3), let $a_1=0$, $a_2=(e)_\rho$. If entering from steps (2.4) or (2.5), let $a_1=a_{1\rho}$, $a_2=a_{2\rho}$. Compute $z' = (0, z_1', z_2', 0) = (a_1', a_2', 0, 0)B_k^{-1}$. If $(z)_j \leqq 0$ for all $j \in I_{B_k}^c$, then QP has no feasible solution. Terminate. If not, compute ν_0^1, ν_0^2 by (3.18) and ν_0, μ by (3.18.3). Return.

Step 5. Let $\theta_{k+1} = \theta_\ell$, $B_{k+1}^{-1} = B_\ell^{-1}$ and update the basic index sets to obtain $J_{B_{k+1}}$, $\bar{J}_{B_{k+1}}^c$, $I_{B_{k+1}}^c$, $\bar{I}_{B_{k+1}}^c$, $I_{B_{k+1}}^s$. Let $k+1 \rightarrow k$ and start the next parametric cycle at step 1.

Remark 3.1

First suppose that θ_ℓ in (3.9) is determined by the fact that exactly one of the variables (3.5.1), (3.5.2) or (3.7.1) - (3.7.3) becomes zero for $\theta = \theta_\ell$. If the pivot element is negative or if, in the case of a zero pivot, the constraint determined by the "Search Procedure", or the "Constraint Replacement Procedure" is unique, then it follows immediately that the pivot operation, or the pair of pivot operations in the case of a zero pivot, gives a basic solution which is valid for some interval $\theta_{\ell+1} \leqq \theta \leqq \theta_\ell$ with $\theta_{\ell+1} < \theta_\ell$.

However, several basic variables of the current RAP may vanish for $\theta = \theta_\ell$. This case can be reduced to the above situation since one can choose the components of d and f in an appropriate way. For instance, if several components of y^c_{1k} and/or y^s_k are zero we can increase all but one of the corresponding components of f and obtain a case with exactly one vanishing component of y^c_{1k} or y^s_k. Similarly, if several components of v_{1k} are zero we can decrease all but one of the corresponding components of d. If several, say n_1 components of x_{1k} vanish, we can choose one of them, perform the pivot operation dictated by the algorithm, and then decrease the component of d corresponding to the new basic variable which then becomes positive. This results in a case with $(n_1 - 1)$ vanishing basic variables. Finally, if several components of n_{1k} are zero, we can choose one of them, perform the necessary pivot step and increase then the component of f corresponding to the new basic variable.

Clearly, any ambiguity caused by combinations of the above cases or by the failure of the "Search Procedure" or the "Constraint Replacement Procedure" to determine a unique constraint can be resolved in a similar way.

Without loss of generality we may therefore assume that Step 2 gives a basic solution, the x-part of which is an optimal solution to PQP for some interval $\theta_{\ell+1} \leqq \theta \leqq \theta_{\ell}$ with $\theta_{\ell+1} < \theta_{\ell}$.

4. COMPUTATIONAL ASPECTS

As indicated in section 3, all operations are performed on the matrix of the "Reduced Auxiliary Problem" (RAP), which is shown in figure 2a for $\theta = \theta_k$ and the corresponding partitioning induced by B_k. Figure 2b depicts the corresponding partitioning of the stand-by constraints. The matrix of the "Auxiliary Problem" (AP) is obtained by considering figures 2a and 2b together. For problems with $m >> n$, for which this method is primarily intended, the size of the working matrix is decisively smaller than the $(m + n)$-order basis commonly used by other algorithms. This reduction is enjoyed, of course, at the expense of the "constraint replacement" operation discussed in section 3. The relative merits of employing this reduced basis depend on the particular structure of the constraint matrix. It is clear however, that for problems with $m \leqq n$ the definition of AP and RAP coincide, and Case II(iii) of the algorithm does not arise.

$$
\begin{array}{cccccccc}
x_{1k}, & x_{2k}, & v_{1k}, & v_{2k}, & u_{1k}^{c}, & u_{2k}^{c}, & y_{1k}^{c}, & y_{2k}^{c}
\end{array}
$$

$$
\begin{bmatrix}
C_{11} & C_{12} & -I & 0 & A_{11}^{c'} & A_{21}^{c'} & 0 & 0 \\
C_{21} & C_{22} & 0 & -I & A_{12}^{c'} & A_{22}^{c'} & 0 & 0 \\
A_{11}^{c} & A_{12}^{c} & 0 & 0 & 0 & 0 & I & 0 \\
A_{21} & A_{22} & 0 & 0 & 0 & 0 & 0 & I
\end{bmatrix}
\begin{array}{l}
= c_1 + \theta d_1 \\
= c_2 + \theta d_2 \\
= b_1^{c} + \theta f_1^{c} \\
= b_2^{c} + \theta f_2^{c}
\end{array}
$$

Figure 2a - Partitioned form of RAP matrix.

$$
\begin{array}{ccccccccc}
x_{1k} & x_{2k} & v_{1k} & v_{2k} & u_{1k}^{c} & u_{2k}^{c} & y_{1k}^{c} & y_{2k}^{c} & y_{k}^{s}
\end{array}
$$

$$
\begin{bmatrix}
A_1^{s} & A_2^{s} & 0 & 0 & 0 & 0 & 0 & 0 & I
\end{bmatrix}
= b^{s} + \theta f^{s}
$$

Figure 2b - Partitioned form of the stand-by constraints.

We now examine three basic computational tools to be used for efficient programming of the proposed algorithm.

Pivot operations:

For all exchange operations the updating of the current inverse is accomplished by the known rules of the revised simplex method. The inverse B_k^{-1} is premultiplied by an "elementary matrix" denoted by E_{k+1}, i.e.

(4.1.1) $$B_{k+1}^{-1} = E_{k+1} B_k^{-1} .$$

If we let (s, p) be the designated pivot position in the matrix of RAP, then we define an "elementary column matrix," E_{k+1}^c , as an identity matrix with its s^{th} column replaced by:

(4.1.2) $$\eta_{k+1}^c = \left(-\frac{(h_p)_1}{(h_p)_s}, \ldots, -\frac{(h_p)_{s-1}}{(h_p)_s}, \frac{1}{(h_p)_s}, -\frac{(h_p)_{s+1}}{(h_p)_s}, \ldots, -\frac{(h_p)_{2n+1}}{(h_p)_s} \right)'$$

with

(4.1.3) $$h_p = B_k^{-1} g_p$$

where g_p is the column of the RAP matrix shown in figure 2a , corresponding to the variable entering the basis. Thus, to perform any variable exchange operation, we identify g_p and apply (4.1.3), (4.1.2) and (4.1.1) with $E_{k+1} = E_{k+1}^c$.

In order to perform the reclassification of a "stand-by" constraint as "current" in Case II(iii-a), we define an "elementary row matrix", denoted by E_{k+1}^r, which is an identity matrix with its σ^{th} row replaced by

$$(4.1.4) \qquad \eta^{r\prime}_{k+1} = (g^{\prime}_{\tau}, \; e^{\prime}_{\tau})$$

where using our previous notation, the p-row vector $g^{\prime}_{B} = (a^{c\prime}_{\tau} - a^{s\prime}_{\rho})$ and $e^{\prime}_{\tau}$ is the $(2n - p + 1)$-row unit vector with the 1 in the position corresponding to $a^{c\prime}_{\tau}$. The row vectors $a^{c\prime}_{\tau}$ and $a^{s\prime}_{\rho}$ are the τ^{th} and ρ^{th} rows of A^{c}_{21} and A^{s}_{1} respectively. Having constructed $\eta^{r\prime}_{k+1}$ we then apply (4.1.1) with $E_{k+1} = E^{r}_{k+1}$. It is easily seen that the effect of this transformation on B^{-1}_{k} is the replacement of the τ^{th} row of A^{c}_{21} by the ρ^{th} row of A^{s}_{1} in B_{k}.

<u>The product form of the inverse</u>.

The operations of exchanging variables or constraint rows are accomplished by successively updating the (2n+1)-order inverse of the current basis. In addition, in several instances during the algorithm, this inverse is used to update a row or column vector (e.g. (3.4), (3.10), (3.16)). These operations may obviously be performed using the explicit form of B^{-1}_{k} and by carrying out (4.1.1) for each exchange as it occurs. However, it is also possible to represent the inverse as the product of elementary matrices of the form E^{c}_{j} or E^{s}_{j}, and rather than performing (4.1.1), the new E_{k+1} may be simply recorded for later use. In practice, only the η^{c}_{j} or η^{r}_{j} are recorded and used in all succeeding computation. The product form of the inverse has been found advantageous in organizing the revised simplex method for large computer systems(2). The economy of storage space and computational effort which is derived from its use is mainly based on the fact that large problems are characterized by sparse arrays. We outline here the tools needed to implement the proposed parametric algorithm using the product form of the inverse.

---

(2) For an excellent account of this method see [3] or [19].

We initiate the algorithm by letting $x_0 = 0$, which implies that $B_0 = B_0^{-1} = I$. Suppose now, that k exchanges have been performed starting from the identity matrix, and that for each exchange the corresponding elementary matrix has been saved. Then, we have

$$B_k^{-1} = E_k \cdot E_{k-1} \cdots E_1 \cdot I \tag{4.2}$$

where E_j may be a column or row elementary matrix. For $m \leqq n$, $E_j = E_j^c$ for all j. However, since in practice $m >> n$ we will probably have $E_j = E_j^r$ for some j. The E_j are saved by recording the type, i.e. c or r, the pivot index, i.e. s or τ, and the vector η_j^c or η_j^r depending on the type specified. Thus, to obtain B_{k+1}^{-1} from B_k^{-1} it suffices to append η_{k+1}^c or η_{k+1}^r to the existing list of elementary vectors. There is no computation of the form (4.1.1); there is merely an accumulation of information in a serial fashion.

The transformation of a column vector g_p (pivot column) by a single E_j is given by:

$$(E_j\, g_p)_i \begin{cases} = (g_p)_i + (\eta_j^c)_i\,(g_p)_s \text{ for } i \neq s, \text{ and } = (\eta_j^c)_i\,(g_p)_s \text{ for } i = s; \\ \qquad \text{when } E_j = E_j^c \\ = (g_p)_i \text{ for } i \neq \tau, \text{ and } = \sum_{\mu=1}^{2n+1} (\eta_j^r)_\mu\,(g_p)_\mu \text{ for } i = \tau; \\ \qquad \text{when } E_j = E_j^r . \end{cases} \tag{4.3}$$

Similarly, the transformation of a row vector g_p' by a single E_j is given by:

$$
(4.4)\qquad (g_p' E_j)_i \begin{cases} = (g_p)_i \text{ for } i \neq s, \text{ and } = \sum_{\mu=1}^{2n+1} (g_p)_\mu (\eta_j^c)_\mu \text{ for } i = s; \\ \qquad\qquad \text{when } E_j = E_j^c . \\ = (g_p)_i + (\eta_j^c)_i (g_p)_\sigma \text{ for } i = \tau, \text{ and } = (\eta_j^c)_i (g_p)_\sigma \text{ for } i = \tau ; \\ \qquad\qquad \text{when } E_j = E_j^r . \end{cases}
$$

We note that the transformation of a row or column vector by an elementary column or row matrix requires the calculation of only one inner product of order (2n+1).

The operation of updating a column vector g_p by the current inverse, e.g. performing (4.1.3), is accomplished by the repeated use of (4.3) to form

$$
(4.5)\qquad h_p = (E_k^c (E_{k-1}^c (\ldots (E_{\ell+1}^c (E_\ell^r (E_{\ell-1}^c (\ldots (E_1^c g_p)\ldots))))\ldots)))
$$

in the "forward" order, i.e. in the order the E_j were generated. Hence, the commonly used term "forward transformation". In the same way, the operation for updating a row vector g_p' by the current inverse is:

$$
(4.6)\qquad h_p' = (\ldots((((\ldots((g_p' E_k^c) E_{k-1}^c)\ldots) E_{\ell+1}^c) E_\ell^r) E_{\ell-1}^c)\ldots) E_1^c
$$

computed in the "backward" order, i.e. in the order opposite from that in which they were generated. Hence the term "backward transformation".

The transformations (4.5), (4.6) are the most important basic tools to be used in carrying out the parametric algorithm, using the product form of the inverse. All of these transformation procedures are integral parts of most commercially available linear programming codes and may be used for our purposes. Slight modifications may be required to handle the case $E_j = E_j^r$ for some of these codes.

Reinversion

As successive parametric steps are performed, the number of recorded elementary transformation vectors (η-vectors) increases. This increase effects the progress of calculations, both from the numerical accuracy and algorithmic efficiency stand points. As the number of the η-vectors increases, so does the effort required to perform a pivot-step, obtain the next parameter value, and apply the "search" or "constraint replacement" procedures, etc. Similarly, the effect of inherent truncation error which accumulates during the forward and backward transformations may reach prohibitive levels. Qualitatively speaking, the main problem here lies in the overall increase in the number of non-zero elements representing the current inverse of the basis.

In such cases, the technique of "reinversion" is employed to re-define the inverse of the current basis in terms of (2n+1) column η-vectors. Recently, reinversion techniques have been given a great deal of attention in connection with large linear programming codes. In particular, the technique of triangularization has been reported to minimize the number of non-zero elements in the η-vectors after reinversion, when applied to sparse matrices. Discussion of this technique and of its many variations is outside the scope of this paper. Nevertheless, such efficient reinversion procedures are available as integral parts of most commercial linear programming codes and may readily be used for our purposes.

5. POST OPTIMALITY ANALYSIS

The optimal solution to a quadratic program of practical interest may not provide the analyst with all the required information for an effective solution of his problem. For example, the cost information and right hand side vector may have some estimated or incorrect elements. It is desirable to investigate the sensitivity of the model to such contingencies. Such information is also useful in the economic interpretation of the dual solution in marginal analysis, since its validity depends on the "ranges" over which the optimal basis remains unaltered. Another instance when such information is required is in short and long range planning, where it is necessary to investigate the nature of the solution by changing the cost and requirements information as linear functions of a parameter.

The "ranges" over which the elements of the given vectors c and b may be altered, without causing a change in the optimal basis, are a ready by-product of the algorithm described in section 3.

Now, for parametric programming, let x_0 be an optimal solution to QP, Δc and Δb be given "change" vectors, and consider the problem:

Maximize

(5.1.1) $$(c + \phi.\Delta c)' x - \frac{1}{2}x' Cx$$

subject to

(5.1.2) $$Ax \leqq b + \phi.\Delta b$$

(5.1.3) $$x \geqq 0$$

where solutions to (5.1) are required for all $0 \leqq \phi \leqq 1$. Construct vectors d and f such that:

(5.2.1) $$(d)_i = -(\Delta c)_i \; ; \; i = 1, \ldots, n,$$

(5.2.2) $$(f)_j = -(\Delta b)_j \; ; \; j = 1, \ldots, m,$$

choose $\theta_0 = 1$ and consider the PQP:

Maximize

(5.3.1) $$((c + \Delta c) + \theta d)' x - \frac{1}{2} x' \, Cx$$

subject to

(5.3.2) $$Ax \leqq (b + \Delta b) + \theta f$$

(5.3.3) $$x \geqq 0$$

for all $0 \leqq \theta \leqq \theta_0$. This is obviously equivalent to considering (5.1) for all $0 \leqq \phi \leqq 1$.

Thus the algorithm of section 3 may be used for parametric programming on the vectors c and b , by a trivial modification of the given data.

6. THEORETICAL JUSTIFICATION OF THE METHOD

In this section the validity of the algorithm will be demonstrated.

All discussions in this section are based on a simplified system representing the RAP basis, which is written as:

$$(6.1) \qquad B = \begin{array}{c} \begin{array}{cc} x\ , & w \end{array} \\ \begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix} \end{array}$$

where in terms of the notation introduced in (3.3)

$$(6.2) \qquad \begin{aligned} C &= \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad ; \quad A_1 = \begin{pmatrix} A^c_{11} & A^c_{12} \\ 0 & -I \end{pmatrix} \quad \text{(active constraints)} \\ x &= (x_1, \quad x_2) \quad ; \quad w = (u^c_1, \quad v_1)\ . \end{aligned}$$

The representation of the basis (6.1) is included in the following partitioning of the RAP matrix where the non-negativity restrictions have been expressed explicitly:

$$(6.3) \qquad \begin{array}{c} \begin{array}{cccc|ccc} x_1\ , & x_2\ , & u^c_1\ , & v & y^n_2\ , & y^c_1\ , & y^n_1 \end{array} \\ \left(\begin{array}{cccc|ccc} C_{11} & C_{12} & A^{c\prime}_{11} & & & & \\ C_{21} & C_{22} & A^{c\prime}_{12} & -I & & & \\ A^c_{11} & A^c_{12} & & & & & \\ \hline 0 & -I & & & I & & \\ A^c_{21} & A^c_{22} & & & & I & \\ -I & 0 & & & & & I \end{array} \right) \end{array}$$

where y_1^n, y_2^n are the slack variables corresponding to the non-negativity restrictions. For a basic solution we have $x_2 = 0$, $y_2^n = 0$ and the dotted portion of (6.3) corresponds to (6.1). However, it also corresponds to (3.3) since the columns of x_2 can be removed with no effect, and the slack variables y_1^c and the inactive rows (A_{21}^c) may be appended with no effect to the basic solution.

In the following discussion the term "constraint" will refer to "non-negativity restrictions" and "ordinary constraints" alike.

Lemma 1

Let $Q(x(\theta)) = (\bar{c} + \theta\bar{d})' x - \frac{1}{2} x' Cx$. Suppose $\begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix}$ is non-singular and

$$(6.4.1) \qquad \begin{aligned} x(\theta) &= d_1 + \theta d_2 \\ w(\theta) &= d_3 + \theta d_4 \end{aligned}$$

is the solution of:

$$(6.4.2) \qquad \begin{aligned} Cx + A_1' w &= \bar{c} + \theta\bar{d} \\ A_1 x &= \bar{b} + \theta\bar{f}_1 \end{aligned}$$

then, for $\theta = \theta_0$

1) $$\left.\frac{\partial Q(x(\theta))}{\partial\theta}\right|_{\theta=\theta_0} = \bar{d}' x_0 + \bar{f}_1' w_0$$

2) $$\left.\frac{\partial^2 Q(x(\theta))}{\partial\theta^2}\right|_{\theta=\theta_0} = \bar{d}' d_2 + \bar{f}_1' d_4$$

<u>where</u> $x_0 = d_1 + \theta_0 d_2$ <u>and</u> $w_0 = d_3 + \theta_0 d_4$.

<u>If</u> C <u>is positive semi-definite, then</u> $\bar{d}' d_2 + \bar{f}_1' d_4 \leqq 0$

<u>Proof:</u> Substituting (6.4.1) into (6.4.2) gives:

$$Cd_1 + \theta C d_2 + A_1' d_3 + \theta A_1' d_4 = \bar{c} + \theta \bar{d}$$

$$A_1 d_1 + \theta A_1 d_2 = \bar{b}_1 + \theta \bar{f}_1$$

for all θ . For $\theta = 0$ we have

$$Cd_1 + A_1' d_3 = \bar{c}$$

$$A_1 d_1 = \bar{b}_1$$

Hence it follows that

$$\theta(C d_2 + A_1' d_4) = \bar{d}$$

(6.4.3)

and $A_1 d_2 = \bar{f}_1$

Now, using $x(\theta)$ from (6.4.1)

$$Q(x(\theta)) = \bar{c}' d_1 - \frac{1}{2} d_1' Cd_1 + (c' d_2 + \bar{d}' d_1 - d_1' Cd_2)\,\theta + (\bar{d}' d_2 - \frac{1}{2} d_2' Cd_2)\,\theta^2$$

and differentiating with respect to θ :

$$\left.\frac{\partial Q(x(\theta))}{\partial Q}\right|_{\theta=\theta_0} = \bar{c}' d_2 + \bar{d}' d_1 + d_1' Cd_2 + (2\bar{d}' d_2 - d_2' Cd_2)\,\theta_0 =$$

$$= \bar{c}' d_2 + \bar{d}' (d_1 + 2d_2\theta_0) - d_2' (Cd_1 + Cd_2\theta_0) =$$

$$= \bar{c}' d_2 + \bar{d}' (d_1 + 2d_2\theta_0) - d_2' (\bar{c} + \theta_0 \bar{d} + A_1' w_0)$$

$$= \bar{d}' (d_1 + \theta_0 d_2) + d_2' A_1' w_0$$

$$= \bar{d}' x_0 + \bar{f}_1' w_0$$

where we have used (6.4.1) - (6.4.3).

Next, differentiating

$$\frac{\partial Q(x(\theta))}{\partial \theta} = \bar{d}' x(\theta) + \bar{f}_1' w(\theta)$$

and using (6.4.1) we obtain

$$\frac{\partial^2 Q(x(\theta))}{\partial \theta} = \frac{\partial}{\partial \theta} (\bar{d}' d_1 + \theta \bar{d}' d_2 + \bar{f}_1' d_3 + \theta \bar{f}_1' d_4)$$

or

$$\left.\frac{\partial^2 Q(x(\theta))}{\partial \theta^2}\right|_{\theta = \theta_0} = \bar{d}' d_2 + \bar{f}_1' d_4 .$$

If C is positive semi-definite $Q(x(\theta))$ is concave and therefore $\bar{d}' d_2 + \bar{f}_1' d_4 \leqq 0$.

Remark 6.1

In order to put the above result in a form useful for proving Lemma 2 we consider the two cases of pivoting below.

I) In Case I, for a particular $\theta = \theta_\ell$, we would like to exchange $(w)_\mu$ and $(y)_\mu$ in the system:

(6.5.1) $$Cx + A_1' w + (w)_\mu a_\mu = c + \theta_\ell d$$

(6.5.2) $$A_1 x = b_1 + \theta_\ell f_1$$

(6.5.3) $$a_\mu' x + (y)_\mu = (b)_\mu + \theta_\ell \cdot (f)_\mu$$

where presently $(y)_\mu = 0$ is nonbasic and $(w)_\mu$ is a basic variable at zero level. The current basic solution of (6.5.1) - (6.5.3) is given by:

$$(6.5.4) \qquad \begin{pmatrix} x \\ w_1 \\ (w)_k \end{pmatrix} = B_k^{-1} \begin{pmatrix} c + \theta_\ell d \\ b_1 + \theta_\ell f_1 \\ (b)_k + \theta_\ell (f)_k \end{pmatrix} - B_k^{-1} e_\mu (y)_\mu$$

where $(y)_\mu = 0$ and

$$B_k = \begin{pmatrix} C & A_1' & a_\mu \\ A_1 & 0 & 0 \\ a_\mu' & 0 & 0 \end{pmatrix}$$

For simplicity (6.5.4) may be written as:

$$(6.5.5) \qquad \begin{pmatrix} x \\ w_1 \\ (w)_\mu \end{pmatrix} = g_1 - h_1 (y)_\mu$$

and in particular

$$(6.5.6) \qquad (w)_\mu = (g_1)_\mu - (h_1)_\mu (y)_\mu$$

where $(h_1)_\mu$, denoting the last component of h_1 , is the "pivot element" in Case I .

Now consider (6.5.1) - (6.5.2) and

$$a_\mu' x = (b)_\mu + \theta_\ell (f)_\mu - (y)_\mu$$

If we let $\bar{c} = c + \theta_\ell d$, $\bar{d} = 0$, $\bar{b}_1' = \left((b_1 + \theta_\ell f_1)', (b)_\mu + \theta_\ell (f)_\mu \right)$,

$\bar{f}_1 = e_\mu$ and $\theta = (y)_\mu$, (6.5.6) and Lemma 1 give

$$(6.5.7) \qquad \frac{\partial^2 Q(x(\theta))}{\partial \theta^2} = (h_1)_\mu \leqq 0 .$$

II) In Case II, for a particular $\theta = \theta_\ell$, we would like to exchange $(y)_\rho$ by $(w)_\rho$ in the system:

$$(6.5.8) \qquad Cx + A_1' w \qquad + (w)_\rho a_\rho = c + \theta_\ell d$$

$$(6.5.9) \qquad A_1 x = b_1 + \theta_\ell f_1$$

$$(6.5.10) \qquad a_\rho' x \qquad + (y)_\rho = (b)_\rho + \theta_\ell (f)_\rho$$

where presently $(w)_\rho = 0$ is non-basic and $(y)_\rho$ is a basic variable at zero level.

The current basic solution of (6.5.8) - (6.5.10) is given by:

$$(6.5.11) \qquad \begin{pmatrix} x \\ w_1 \\ (y)_\rho \end{pmatrix} = \hat{B}_k^{-1} \begin{pmatrix} c + \theta_\ell d \\ b_1 + \theta_\ell f_1 \\ (b)_\rho + \theta_\ell (f)_\rho \end{pmatrix} - \hat{B}_k^{-1} \begin{pmatrix} a_\rho \\ 0 \\ 0 \end{pmatrix} (w)_\rho$$

where $(w)_\rho = 0$ and

$$\hat{B}_k = \begin{pmatrix} C & A_1' & 0 \\ A_1 & 0 & 0 \\ a_\rho' & 0 & 1 \end{pmatrix}$$

For simplicity, (6.5.11) may be written as

$$(6.5.12) \qquad \begin{pmatrix} x \\ w_1 \\ (y)_\rho \end{pmatrix} = g_2 - h_2 (w)_\rho$$

and in particular

(6.5.13) $$(y)_\rho = (g_2)_\rho - (h_2)_\rho (w)_\rho$$

where $(h_2)_\rho$, denoting the last component of h_2, is the "pivot element" in Case II.

Lemma 2

Consider Cases I and II of pivotal operations and the corresponding systems (6.5.1) - (6.5.3) and (6.5.8) - (6.5.10). Then, the following properties hold:

A) The pivot element in both Cases I and II is non-positive.

B) If, in Case I, the pivot element is zero and s_1 and s_2 consist of the first n components of g_1 and h_1 respectively as defined by (6.5.5), then, $Q(s_1 - \lambda s_2)$ is either constant or a linear function of λ.

If, in Case II, the pivot element is zero, then, the constraint $a'_\rho x = (b)_\rho + \theta_\ell (f)_\rho$ is linearly dependent on the constraints (6.5.9).

Proof:

Case I: The assertion follows immediately from (6.5.7).

Case II: It follows from the definition of $\hat{B}_k$ and (6.5.11) - (6.5.13) that

(6.6.1) $$(h_2)_\rho = -(a'_\rho, 0) \begin{pmatrix} C & A'_1 \\ A_1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} a_\rho \\ 0 \end{pmatrix}$$

Let $(\bar{v}, \bar{z})$ be the unique solution of the system:

(6.6.2)
(6.6.3) $$\begin{pmatrix} C & A'_1 \\ A_1 & 0 \end{pmatrix} \begin{pmatrix} v \\ z \end{pmatrix} = \begin{pmatrix} a_\rho \\ 0 \end{pmatrix} ,$$

Substituting $(\overline{v}, \overline{z})$ into (6.6.1), premultiplying (6.6.2) by $\overline{v}'$ and using (6.6.3) gives

$$(h_2)_\rho = -(a'_\rho, 0) \begin{pmatrix} \overline{v} \\ \overline{z} \end{pmatrix} = -\overline{v}' C \overline{v} .$$

Since C is positive semi-definite this implies $(h_2)_\rho \leqq 0$. If $\overline{v}' C \overline{v} = 0$, then $C\overline{v} = 0$ (See Lemma 1 in [18]), which, by (6.6.2), means $A_1' \overline{z} = a_\rho$.

<u>Lemma 3</u>

<u>Suppose the pivot element in Case I is zero, i.e.,</u> $(h_1)_\mu = 0$, <u>for</u> $\theta = \theta_\ell$. <u>Let</u> $(\overline{x}_k, \overline{w}_k, (\overline{w}_k)_\mu)$ <u>be the solution of the RAP for</u> $\theta = \theta_\ell$ <u>and</u> $Q(\overline{x}_k)$ <u>the corresponding value of the objective function of PQP. Furthermore, let</u> $\overline{s}_k$ <u>be the vector defined by the "Search Procedure". Then,</u>

<u>A)</u> $Q(\overline{x}_k + \lambda \overline{s}_k) = Q(\overline{x}_k)$ <u>for all</u> $\lambda \in E^1$.

<u>B)</u> $\overline{w}_k(\lambda) = \overline{w}_k$ <u>for all</u> $\lambda \in E^1$.

<u>Proof:</u>

<u>A)</u> A review of the definition (3.10.2) of $\overline{s}_k$ shows that, using the more explicit formulation (6.5.1) - (6.5.3) of RAP we can express $(\overline{x}_k + \lambda \overline{s}_k)$ in the form $(s_1 - \lambda s_2)$, where s_1 and s_2 consist of the first n components of g_1 and h_1 respectively as defined by (6.5.5). Hence, Lemma 2 implies that $Q(\overline{x}_k + \lambda \overline{s}_k)$ is either constant or a linear function of λ.

$A_1 (\overline{x}_k + \lambda \overline{s}_k) = b_1 + \theta_\ell f_1$ for all $\lambda \in E^1$ implies $A_1 \overline{s}_k = 0$. Hence, using RAP (in particular (6.5.1)), we obtain:

$$(c + \theta_\ell d - C\overline{x}_k)' \overline{s}_k = (A_1' w_k)' \overline{s}_k + (w_k)_\mu a'_\mu \overline{s}_k = 0$$

since $(w_k)_\mu = 0$. This proves the first assertion.

B) Since $Q(\overline{x}_k + \lambda \overline{s}_k) = Q(\overline{x}_k)$, we have $\overline{s}_k' C \overline{s}_k = 0$ which implies $C\overline{s}_k = 0$. Hence $C(\overline{x}_k + \lambda \overline{s}_k) = C\overline{x}_k$ for all λ , and since the columns of A_1' are linearly independent, it follows from RAP that $\overline{w}_k$ is independent of λ.

Theorem 1

Let $\overline{x}_k = x_k(\theta_\ell)$ be an optimal solution to PQP for $\theta_\ell > 0$. Suppose, for $\theta = \theta_\ell$, Case I applies and the pivot element is zero.

If the "Search Procedure" yields $\lambda = -\infty$, then the original problem QP has no optimal solution.

Proof

The assertion is clear, if PQ has no feasible solution. Let x_0 be an arbitrary feasible solution of PQ. By assumption,

$$A(\overline{x}_k + \lambda \overline{s}_k) \leqq b + \theta_\ell f \quad \text{and} \quad \overline{x}_k + \lambda \overline{s}_k \geqq 0$$

hold for all $\lambda \leqq 0$. Therefore, $A\overline{s}_k \geqq 0$ and $\overline{s}_k \leqq 0$. Thus, $Ax_0 \leqq b$ and $x_0 \geqq 0$ imply

(6.7.1) $$A(x_0 + \lambda \overline{s}_k) \leqq b \quad \text{and} \quad x_0 + \lambda \overline{s}_k \geqq 0 \quad \text{for all } \lambda \leqq 0 .$$

From the definition of $\overline{s}_k$ and from Lemma 1 it follows that for $\theta = \theta_\ell$

$$\left.\frac{\partial Q(\overline{x}_k + \lambda \overline{s}_k)}{\partial \lambda}\right|_{\lambda=0} = (c + \theta_\ell d - C\overline{x}_k)' \, \overline{s}_k = (w(\theta_\ell))_\mu = 0$$

By the assumption of Case I we have that $(w)_\mu < 0$ for $\theta < \theta_\ell$. Hence, (6.7.2) $$(c' - Cx_k(0))' \bar{s}_k = (w(0))_\mu < 0$$

Since by Lemma 3, for $\theta = \theta_\ell$,

$$\frac{\partial^2 Q(\bar{x}_k + \lambda \bar{s}_k)}{\partial \lambda^2} = 0 ,$$

it follows that $\bar{s}'_k \ C \bar{s}_k = 0$. Hence, for $\theta = 0$ we have

(6.7.3) $$Q(x_0 + \lambda \bar{s}_k) = c' x_0 - \frac{1}{2} x'_0 \ C x_0 + (c - Cx_0)' \bar{s}_k \lambda$$
$$= c' x_0 - \frac{1}{2} x'_0 C x_0 + c' \bar{s}_k \lambda$$

since $\bar{s}'_k C \bar{s}_k = 0$ implies $C \bar{s}_k = 0$ (Lemma 1 in [18]). If $C \bar{s}_k = 0$, it follows from (6.7.2) that $c' \bar{s}_k < 0$. But then (6.7.1) and (6.7.3) show that the objective function of QP can be made arbitrarily large over the feasible domain.

Theorem 2

Let the designated pivot element be zero for some $\theta = \theta_\ell$. Consider the following cases:

A) Let the designated pivot step correspond to Case I, and assume that the "Search Procedure" terminates with a finite $\lambda < 0$ corresponding to a constraint which is to become active.

B) Let the designated pivot step correspond to Case II and assume that the "Constraint Replacement Procedure" yields a presently active constraint to become inactive.

Then, the exchanges of active and inactive constraints prescribed by the "Search Procedure" or by the "Constraint Replacement Procedure" can be accomplished by a pair of pivot steps.

Proof:

A) Consider the current form of RAP :

$$
(6.8.1)\quad
\begin{aligned}
Cx + A_1'w + a_1\omega_1 + a_2\omega_2 &= c + \theta_\ell d \\
A_1x &= b_1 + \theta_\ell f_1 \\
a_1'x + (y)_1 &= \alpha_1 + \theta_\ell \alpha_2 \\
a_2'x + (y)_2 &= \beta_1 + \theta_\ell \beta_2
\end{aligned}
$$

$$\omega_1 \cdot (y)_1 = 0 \;;\; \omega_2 \cdot (y)_2 = 0$$

where a_1 is the constraint to become inactive (Case I) and a_2 is the constraint found by the "Search Procedure" to become active. Since the pivot is zero ω_1 and $(y)_1$ could not be exchanged. Then, the "Search Procedure" was applied which indicated that $(y)_2$ should leave and ω_2 should enter the basis. We would like to show that it is always possible to exchange ω_1 and ω_2 and then $(y)_2$ and $(y)_1$. Thus, we first show that the non-singular basis matrix

$$
(6.8.2)\quad
\begin{array}{cccc}
x, & w, & \omega_1, & (y)_2 \\
\hline
\end{array}
$$

$$
\left(\begin{array}{cccc}
C & A_1' & a_1 & 0 \\
A_1 & 0 & 0 & 0 \\
\hdashline
a_1' & 0 & 0 & 0 \\
a_2' & 0 & 0 & 1
\end{array}\right)
$$

$$\text{①} \quad \text{②} \quad \text{③} \quad \text{④}$$

remains non-singular when a_1 is replaced by a_2 in the column corresponding to ω_1, i.e. in

(6.8.3)
$$\begin{array}{cccc} x\ , & w\ , & \omega_2\ , & (y)_2 \\ \hline \end{array}$$
$$\left(\begin{array}{cccc} C & A_1' & a_2 & 0 \\ A_1 & 0 & 0 & 0 \\ \hline a_1' & 0 & 0 & 0 \\ a_2' & 0 & 0 & 1 \end{array}\right)$$
$$\textcircled{1} \quad \textcircled{2} \quad \textcircled{3} \quad \textcircled{4}$$

column ③ is not a linear combination of the other columns. To this end, we recall that according to the "Search Procedure":

$$(6.8.4) \qquad \begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix} \begin{pmatrix} x_k - \lambda \bar{s}_k \\ w_k - \lambda \tilde{s}_k \end{pmatrix} = \begin{pmatrix} c + \theta_\ell d \\ b_1 + \theta_\ell f_1 \end{pmatrix} \qquad \text{for all } \lambda < 0$$

where $s_k' = (\bar{s}_k', \hat{s}_k')$ consists of the appropriate components of h_1 as defined by (6.5.5). Obviously s_k is in the null space of this matrix, i.e.

$$(6.8.5) \qquad \begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix} \begin{pmatrix} \bar{s}_k \\ \tilde{s}_k \end{pmatrix} = 0$$

Furthermore, since, by assumption, the "Search Procedure" terminated finitely with $a_2'x$ as the first constraint to be encountered we know that:

$$a_2' \bar{s}_k \neq 0 \ .$$

Clearly then, the row $(a_2', 0)$ is not a linear combination of the rows of $\begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix}$.

On the other hand, since (6.8.2) is a basis, the entire columns (1), (2) and (4) of (6.8.3) are linearly independent and (6.8.3) is non-singular if and only if column (3) is not a linear combination of the remaining ones. We have already shown the latter to be true for the upper part of the partitioning in (6.8.3). It is obviously true for the entire (6.8.3). Then, the new basis matrix is non-singular which implies that the pivot step for exchanging ω_1 and ω_2 is possible. Thus ω_2 enters the basis at zero level.

To show that the second pivot, i.e. the exchange of $(y)_2$ and $(y)_1$, is possible we consider the basis matrix (6.8.3) where we would like to replace the column designated as (4), by another unit vector corresponding to $(y)_1$, and have the new basis matrix

$$(6.8.6) \qquad \begin{array}{c} \begin{matrix} x\ , & w\ , & \omega_2\ , & (y)_1 \end{matrix} \\ \left(\begin{array}{cccc} C & A_1' & a_2 & 0 \\ A_1 & 0 & 0 & 0 \\ \hline a_1' & 0 & 0 & 1 \\ a_2' & 0 & 0 & 0 \end{array}\right) \\ \begin{matrix} (1) & (2) & (3) & (4') \end{matrix} \end{array}$$

be non-singular. Since $\begin{pmatrix} a_2 \\ 0 \end{pmatrix}$ is not linearly dependent on the columns of $\begin{pmatrix} C & A_1' \\ A_1 & 0 \end{pmatrix}$ it follows that (4') cannot be written as a linear combination of the columns in (1), (2) and (3). Thus (6.8.6) is non-singular, and the exchange of $(y)_2$ by $(y)_1$ is possible. **Clearly**, $(y)_1$ enters the basis at a positive level.

In order to show that ω_2 becomes positive for $\theta < \theta_\ell$, let $(\bar{x}(\theta), \bar{w}(\theta), \omega_1(\theta), y_2(\theta))$ and $(\hat{x}(\theta), \hat{w}(\theta), \omega_2(\theta), y_1(\theta))$ denote the basic solution of (6.8.1) before and after the pair of pivot steps has been performed, respectively. Let $\bar{s}_k$ be the vector determined by the "Search Procedure". Then, considering (6.8.1) with $x = \bar{x}_k + \lambda \bar{s}_k$ and $\theta = \theta_\ell$ it can easily be shown that:

$$A_1 \bar{s}_k = 0 , \quad a_1' \bar{s}_k > 0 , \text{ and } a_2' \bar{s}_k < 0 .$$

The first relation of (6.8.1) for the first basic solution, post-multiplied by $\bar{s}_k$, is:

$$(c + \theta d - C\bar{x}(\theta))' \bar{s}_k = (A_1' \bar{w}(\theta))' \bar{s}_k + \omega_1(\theta) a_1' \bar{s}_k$$

It follows that, for $\theta < \theta_\ell$,

$$(c + \theta d)' \bar{s}_k < 0$$

since by the proof of Theorem 1, $C \bar{s}_k = 0$ and since furthermore $\omega_1(\theta) < 0$ for $\theta < \theta_\ell$ by the assumptions of Case I.

Similarly, from the first relation of (6.8.1) for the second basic solution:

$$(c + \theta d - C\hat{x}(\theta))' \bar{s}_k = (A_1' \hat{w}(\theta))' \bar{s}_k + \omega_2(\theta) a_2' \bar{s}_k$$

for $\theta < \theta_\ell$,

$$\omega_2(\theta) a_2' \bar{s}_k = (c + \theta d)' \bar{s}_k < 0$$

But since $a_2' \bar{s}_k < 0$, we must have $\omega_2(\theta) > 0$ for $\theta < \theta_\ell$.

B) Consider again (6.8.1) and assume that a_2 corresponds to the constraint to become active (Case II). Since the pivot is zero, $(y)_2$ and ω_2 could not be exchanged. Suppose the "Constraint Replacement Procedure" determined the constraint corresponding to a_1 to become inactive. By arguments similar to those used under part A above it can be shown that first ω_1 and ω_2 and then $(y)_2$ and $(y)_1$ can be exchanged by a pivot operation. Lemma 4 below assures that for $\theta < \theta_\ell$, the new basic variables ω_2 and $(y)_1$ are non-negative.

Lemma 4[3)]

Let C be a (n, n)-matrix, B a (m, n)-matrix whose rows are linearly independent. Consider the equations

$$\begin{aligned} Cx + B'u &= c \\ Bx &= b + \theta f , \end{aligned}$$

where c is an n-vector while b and f are m-vectors and $\theta \geqq 0$ is a scalar.

Denote the inverse of the matrix

$$\begin{pmatrix} C & B' \\ B & 0 \end{pmatrix} \text{ by } \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} \text{ where } M_3 = M_2' .$$

Suppose, for $\theta_0 > 0$, the above equations have a solution $(x_0(\theta), u_0(\theta))$ which has the following properties :

$$\begin{aligned} & u_0(\theta_0) \geqq 0 \\ (6.9.1) \qquad & d' x_0(\theta_0) = \alpha + \theta_0 \beta \\ & d' x_0(\theta) < \alpha + \theta\beta \quad \text{for} \quad \theta > \theta_0 \end{aligned}$$

3) The proof of this Lemma may be found in [16].

where the n-vector d is a linear combination of the columns of B', i.e., $d = B'z$, while α and β are scalars.

1) If $M_3 d \leqq 0$ the inequalities

$$Bx \leqq b + \theta f$$

$$d'x \leqq \alpha + \theta\beta$$

are inconsistent for $\theta < \theta_0$.

2) Suppose $M_3 d$ has at least one positive component. Let, for $\theta = \theta_0$,

$$\mu_0 = \frac{(u_0)_k}{(M_3 d)_k} = \min \left\{ \frac{(u_0)_j}{(M_3 d)_j} \quad \text{for all } j \text{ with } (M_3 d)_j > 0 \right\}.$$

Replace the kth rows of B by d' and denote the new matrix by $\bar{B}$. Furthermore, replace the kth component of b and f by α and β, respectively, and denote the new vectors by $\bar{b}$ and $\bar{f}$.

Then the columns of $\bar{B}$ are linearly independent and the system

$$\begin{aligned} Cx + \bar{B}'u &= c \\ \bar{B}x &= \bar{b} + \theta\bar{f} \end{aligned} \qquad (6.9.2)$$

has a solution $(x_1(\theta), u_1(\theta))$ with the properties

$$u_1(\theta_0) \geqq 0$$

$$x_1(\theta_0) = x_0(\theta_0)$$

$$b_k' x_1(\theta) < (b)_k + \theta(f)_k \qquad \text{for } \theta < \theta_0$$

where b_k denotes the kth row of B.

Remark 2

Theorem 1 and Lemma 4 state an important property of the algorithm described in section 3, which allows the detection of off-normal conditions present in QP or PQP by simple tests to be performed on RAP.

Thus, if, for some $\theta = \theta_\ell$, the pivot in Case I is zero and $\underline{\lambda} = -\infty$, it is immediately concluded that no optimal solution to PQP , for $\theta < \theta_\ell$, and to QP exists. If, however, it is known a priori that QP has a non-empty feasible domain, then the conclusion is that PQP, for $\theta < \theta_\ell$, and QP have unbounded solutions over their respective feasible domains. As a further clarification, consider the case of an unbounded feasible domain and an objective function whose value is bounded from above on this domain. Then, the "Search Procedure" will yield a finite $\underline{\lambda}$ and the normal course of the algorithm is followed.

The existence of an empty feasible domain in PQP for $\theta < \theta_\ell$, and hence for QP, is detected from RAP by a zero pivot element in Case II and the failure of the "Constraint Replacement Procedure" evidenced by (3.17) or by part 1 of Lemma 4 .

Theorem 3

The algorithm outlined in section 3 gives one of the following alternatives after a finite number of steps:

a) an $\bar{x}_k = x_k(0)$ which is an optimal solution to QP ,

b) the information that QP has no optimal solution.

Proof:

In view of Remark 3.1 we may assume that at each parametric step, we obtain an interval $[\theta_{k+1}, \theta_k]$ with $\theta_{k+1} < \theta_k$ such that $x_k(\theta) = (x_{1k}, 0)$ with x_{1k} given by (3.4.2) is the optimal solution of PQP for $\theta \in [\theta_{k+1}, \theta_k]$. The representation of $x_k(\theta)$ as function of θ is uniquely determined by the set of active constraints. Therefore, there is only a finite number of different functions $x_j(\theta)$ which represent the optimal solution to PQP for some interval $[\theta_{j+1}, \theta_j]$. Hence, the algorithm described in section 3 gives, after a finite number of steps, an interval $[\theta_{k+1}, \theta_k]$ and a corresponding optimal solution $x_k(\theta)$ to PQP such that either $0 \in [\theta_{k+1}, \theta_k]$ or the algorithm fails for $\theta < \theta_{k-1}$. In the first case, $x_k(0)$ is an optimal solution to QP. The second case occurs if the pivot element is zero. Then, depending on which case applies, either the "Search Procedure" gives $\underline{\lambda} = -\infty$ or the "Constraint Replacement Procedure" is unable to determine a constraint to become inactive. Both alternatives indicate that QP has no optimal solution (Theorem 1, Lemma 4).

REFERENCES

1. Barankin, E. W. and Dorfman, R.: "On Quadratic Programming", University of California Publications in Statistics, No. 2, pp. 285-318, 1958.

2. Beale, E. M. L.: "On Quadratic Programming," Naval Logistics Research Quarterly, Vol. 6, pp. 227-244, 1959.

3. Beale, E. M. L.: "Numerical Methods," Non-Linear Programming (J. Abadie, Ed.), pp. 143-170, North Holland Publishing Co., Amsterdam, 1967.

4. Boot, J. C. G., "Quadratic Programming, Algorithms-Anomalies-Application," North Holland Publishing Company, Amsterdam, 1964.

5. Dantzig, G. B., "Quadratic Programming, A Variant of the Wolfe-Markowitz Algorithms," Operations Research Center Report #2, University of California, Berkeley, California, 1961.

6. Dantzig, G.: "Linear Programming and Extensions," Princeton University Press, Princeton, New Jersey, 1963.

7. Dorn, W. S., "Non-Linear Programming - A Survey," Management Science, Vol. 9, pp. 171-208, 1963.

8. Kuhn, H. W. and A. W. Tucker, "Non-Linear Programming," Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley and Los Angeles, pp. 481-492, 1951.

9. Lemke, C. E., "A Method of Solution for Quadratic Programs," Management Science, Vol. 8, No. 4, pp. 442-453, 1962.

10. Markowitz, H. M., "Portfolio Selection," Cowles Foundation Monograph, No. 16, John Wiley and Sons, Inc., New York, 1959.

11. Moore, J. H., Whinston, A. B., "Experimental Methods in Quadratic Programming," Management Science, Vol. 13, pp. 58-76, 1966.

12. Panne, C, Van de and Whinston A., "The Simplex and the Dual Method for Quadratic Programming," Econ. Inst. of the Netherlands School of Economics, Report #6314, Rotterdam, 1962.

13. Ritter, K., "Ein Verfahren zur Lösung parameter-abhängiger, nichtlinearer Maximum-Probleme," Unternehmensforschung, Vol.6, pp. 149-146, 1962. (Also translation in Naval Logistics Research Quarterly, Vol. 14, pp. 147-162, 1967).

14. Ritter, K., "Stationary Points of Quadratic Maximum Problems," Z. Wahrscheinlichkeitstheorie verw. Gebiete Vol. 4, pp. 149-158, 1965.

15. Ritter, K., "A Parametric Method for Solving Certain Non-Concave Maximization Problems," Journal of Comp. and System Sciences, Vol. 1, pp. 44-54, 1967.

16. Ritter, K., "A Decomposition Method for Structured Quadratic Programming Problems," Journal of Comp. and System Sciences, Vol. 1, pp. 241-260.

17. Theil, H. and Van de Panne, C., "Quadratic Programming as an Extension of Conventional Quadratic Maximization," Management Science, Vol. 7, pp. 1-20, 1960.

18. Wolfe, P., "The Simplex Method for Quadratic Programming," Econometrica, Vol. 27, pp. 382-398, 1959.

19. Wolfe, P., "The Product Form of the Simplex Method," Non-Linear Programming (J. Abadie, Ed.) pp. 305-309, North Holland Publishing Co., Amsterdam, 1967.

20. Zoutendijk, G., "Methods of Feasible Directions", Elsavier Publishing Co., Amsterdam, 1960.