NEAREST POINTS IN NONSIMPLICIAL CONES AND LCP'S WITH PSD SYMMETRIC MATRICES

K.S. AL SULTAN
Department of Systems Engineering
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

and

K.G. MURTY

Department of
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 90-22

August 1990

NEAREST POINTS IN NONSIMPLICIAL CONES AND LCP'S WITH PSD SYMMETRIC MATRICES

K.S. AL-SULTAN and K.G. MURTY
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117, U.S.A.
and
Department of Systems Engineering
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

Abstract We discuss conditions for the equivalence of a linear complementary problem (LCP) with a positive semidefinite (PSD) symmetric matrix to a nearest point problem over a Pos cone. We then develop an algorithm for this nearest point problem, and report on its computational performance.

Key Words: nearest point problem, convex polyhedral cones, Pos cones, linear complementary problem.

1. INTRODUCTION

Given a point $q \in \mathbf{R^n}$ and a matrix Q of order $n \times m$, we consider the nearest point problem (NPP), denoted by the symbol $[Q, q]$, of finding the nearest point (by Euclidean distance) to q in the cone $\text{Pos}(Q) = \{x : x = Q\lambda, \lambda = (\lambda_1, ..., \lambda_m)^T \geq 0\}$. A variety of applications of this problem in robotics have been discussed[2]. This problem also appears prominently in remote sensing where the composition of a mixture of various constituents is to be estimated using measurements on signals from the mixture and from the pure constituents. D.R. Wilhelmsen[8] discusses applications of the NPP in constructing approximations of functions. Finally, the direction finding routine in each Step of the gravitational method for linear programs[1] is an NPP. We extend an algorithm developed[7] for the case of a simplicial cone, to the NPP considered here.

For any matrix A, we denote by $A_{i.}$, $A_{.j}$, its ith row, jth column respectively. If $\mathbf{S}$ and $\mathbf{T}$ are two sets $\mathbf{S}\backslash\mathbf{T}$ denotes the set of all elements of $\mathbf{S}$ which are not in $\mathbf{T}$. $|\mathbf{S}|$ denotes the cardinality of the set $\mathbf{S}$. For any vector x, $\|x\|$ denotes its Euclidean norm.

Without any loss of generality we assume that every column vector of Q is nonzero. The problem is to find $\lambda = (\lambda_1, ..., \lambda_m)^T$ to

$$\begin{aligned} &\text{minimize } \|q - Q\lambda\|^2 \\ &\text{subject to } \lambda \geq 0 \end{aligned} \tag{1}$$

An optimum solution λ^* for (1) is called an *optimum combination vector* for the NPP $[Q; q]$, then the desired nearest point in $\text{Pos}(Q)$ is $x^* = Q\lambda^*$. The nearest point x^* is always unique, but the optimum combination vector λ^* may not be unique.

If rank Q is $p < n$ obtain any maximal linearly independent subset of rows of Q, say $\{Q_{1.}, \cdots, Q_{p.}\}$, let Q' be the $p \times m$ submatrix of Q consisting of these rows, and $\mathbf{F}$ their linear hull. Let $\bar{q} = (\bar{q}_1, \cdots, \bar{q}_n)^T$ be the orthogonal projection of q in $\mathbf{F}$, and let $q' = (\bar{q}_1, \cdots, \bar{q}_p)$ be the subvector of $\bar{q}$ corresponding to the rows in Q'. Then any optimum combination vector for $[Q; q]$ is also an optimum combination vector for $[Q'; q']$ and conversely. Thus, it is possible to transform the NPP $[Q; q]$ into one of the same form with the matrix of full row rank, however this transformation is computationally expensive, as it involves finding a maximal linearly independent subset of rows of the original Q. Our algorithm works whether Q is of full rank or not.

For any $1 \leq r \leq m$, if the ray of $Q_{.r}$ is not an extreme ray of $\text{Pos}(Q)$, deleting the column $Q_{.r}$ from Q does not change the cone $\text{Pos}(Q)$, and hence the NPP, and this simplifies the NPP. However, detecting such $Q_{.r}$ is computationally expensive. Our algorithm will work even if some $Q_{.r}$ may correspond to non-extreme rays in $\text{Pos}(Q)$.

2. THE SPECIAL CASE OF DIMENSION 2

Suppose $n = 2$. The following direct method can be used.

Step 1: In this case it is possible to find a feasible solution, λ^*, to the $2 \times m$ system $Q\lambda = q$, $\lambda \geq 0$, if one exists, by simple geometric procedures. Such a λ^* is an optimum combination vector for $[Q; q]$. Otherwise, there is no feasible solution to this system, go to Step 2.

Step 2: For $1 \leq j \leq m$, the nearest point, q^j in the ray of $Q_{.j}$ to q is given by

$$q^j = \begin{cases} 0 & \text{if } (Q_{.j})^T q \leq 0 \\ Q_{.j}\left(\frac{(Q_{.j})^T q}{\|Q_{.j}\|^2}\right) & \text{if } (Q_{.j})^T q > 0 \end{cases} \tag{2}$$

Let ρ be such that q^ρ is the closest among all the q^j to q. Define $\lambda^* = (\lambda_j^*)$ by $\lambda_j^* = 0$ for all $j \neq \rho$, and $= (Q_{.\rho})^T \frac{q}{\|Q_{.\rho}\|^2}$ for $j = \rho$. This λ^* is the optimum combination vector for $[Q; q]$.

3. RELATIONSHIPS TO LCP'S WITH SYMMETRIC PSD MATRICES

The KKT optimality conditions for (1) lead to the following LCP: find $\mu = (\mu_1, ..., \mu_m)^T$, λ, satisfying

$$\begin{aligned} &\mu - (Q^T Q)\lambda = -Q^T q \\ &\mu \geq 0, \quad \lambda \geq 0 \ , \quad \mu^T \lambda = 0 \end{aligned} \tag{3}$$

If $(\bar{\mu}, \bar{\lambda})$ is a solution of the LCP (3), then $\bar{\lambda}$ is an optimum solution for (1). Conversely, if $\bar{\lambda}$ is an optimum solution of (1), let $\bar{\mu} = Q^T Q\bar{\lambda} - Q^T q$, then $(\bar{\mu}, \bar{\lambda})$ is a solution of the LCP (3). Thus, any NPP is equivalent to an LCP with a PSD symmetric matrix.

Now, suppose M is a PSD symmetric matrix of order $m \times m$ and rank n. Consider the LCP (b, M), which is to find $w \in \mathbf{R}^m$, $z \in \mathbf{R}^m$ satisfying

$$\begin{aligned} &w - Mz = b \\ &w, z \geq 0 \ , \quad w^T z = 0 \end{aligned} \tag{4}$$

We can find a matrix Q of order $n \times m$ and rank n such that $M = Q^T Q$ (for example, Q can be taken as the Cholesky factor of M. Subroutine LCHRG in IMSL computes this). Consider the following system in variables $y = (y_1, ..., y_n)^T$

$$Q^T y = -b \tag{5}$$

(5) may not have a solution. As an example, consider

$$b = \begin{pmatrix} 4 \\ 7 \end{pmatrix} \ , \quad M = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Here $Q = (1, 1)$. In this example it can be verified that system (5) has no solution, even though the LCP (b, M) has the unique solution $(w = (3, 0)^T, z = (0, 7)^T)$.

If (5) does have a solution y, then the LCP (b, M) is equivalent to the NPP $[Q; y]$ as discussed above.

Theorem 1: Consider the LCP (b, M) where M is a PSD symmetric matrix of order m and rank n. Let $\hat{Q}$ be any matrix of order $n \times m$ satisfying $M = \hat{Q}^T\hat{Q}$. Then the LCP (b, M) can be transformed into a nearest point problem iff the system $\hat{Q}^T y = -b$ has a solution y.

Proof: If y is a solution of $\hat{Q}^T y = -b$, the LCP (b, M) is equivalent to the NPP $[\hat{Q}; y]$ as discussed above.

Now, suppose LCP (b, M) is equivalent to an NPP $[\tilde{Q}; \tilde{q}]$. Without any loss of generality we can assume that $\tilde{Q}$ is of full row rank. From the above discussion we know that $\tilde{Q}^T\tilde{Q} = M$ and $\tilde{Q}^T\tilde{q} = -b$. Since M is of order $m \times m$ and rank n, these facts imply that $\tilde{Q}$ is of order $n \times m$. From standard results in linear algebra we know that if Q is any matrix of order $n \times m$ satisfying $Q^TQ = M$, then the linear hull of the column vectors of Q^T is the same as $\mathbf{F}$, the linear hull of the column vectors of M, and hence (5) has a solution iff $-b \in \mathbf{F}$ which depends only on M. Thus, if (5) has a solution y for some Q satisfying $Q^TQ = M$, then it has a solution for all such Q. These facts imply that the system $\hat{Q}^T y = -b$ must have a solution y too, in this case.

∎

Corollary 1: The LCP (b, M) where M is a PSD symmetric matrix, can be transformed into an NPP iff $b \in \mathbf{F}$, the linear hull of the set of column vectors of M.

Proof: Follows from the proof of Theorem 1.

∎

A vector $\bar{\lambda} \in \mathbf{R^m}$, $\bar{\lambda} \geq 0$, is an optimum solution for (1) iff there exists a $\bar{\mu} \in \mathbf{R^m}$, $\bar{\mu} \geq 0$, such that $(\bar{\mu}, \bar{\lambda})$ together satisfy (3), hence $\bar{\mu} = Q^T(Q\bar{\lambda}) - Q^Tq$, and as $Q\bar{\lambda}$ is constant for all optimum solutions $\bar{\lambda}$ to (1), this implies that the vector μ remains the same in all solutions of the LCP (3). Thus, if $(\bar{\mu}, \bar{\lambda})$ is a solution of the LCP (3), every solution of it is of the form $(\bar{\mu}, \lambda)$ where λ satisfies: $\lambda_j = 0$ if j is such that $\bar{\mu}_j > 0$, $\bar{\mu} - Q^TQ\lambda = -Q^Tq$, and $\lambda \geq 0$.

Definition: An index j, $1 \leq j \leq m$, is said to be a *critical index* for the LCP (3), or for the NPP (1), if $\lambda_j > 0$ in some solution of (3),

or equivalently, if the nearest point in Pos(Q) to q can be expressed as $\sum_{k=1}^{m} \bar{\lambda}_k Q_{.k}$ where $\bar{\lambda}_k \geq 0$ for all k, and > 0 for $k = j$.

Hence, if j is a critical index for (3), (1), $\mu_j = 0$ in every solution for (3).

Reduction to a lower dimensional problem using a critical index

Consider (1), (3). Let $M = (m_{ij}) = Q^T Q$, $b = (b_i) = -Q^T q$. Suppose (1) is a critical index. Then in every solution to (3) we have $\mu_1 = 0$, and hence (3) is equivalent to

$$\begin{aligned} & -M_{1.}\lambda = b_1 \\ & \mu_i - M_{i.}\lambda = b_i \ , \quad i = 2 \ \text{ to } \ m \\ & \mu_2 \text{ to } \mu_m \geq 0 \ , \quad \lambda \geq 0 \ ; \quad \text{and} \quad \mu_i \lambda_i = 0 \ , \quad i = 2 \text{ to } m \ . \end{aligned} \tag{6}$$

Since $m_{11} = (Q_{.1})^T Q_{.1} > 0$, from the first equation in (6), we can get λ_1 in terms of λ_2 to λ_m and eliminate it from the system. This leads directly into an LCP in the variables μ_2 to μ_m and λ_2 to λ_m. To get this lower order LCP perform a Gaussian Pivot Step on $(-M \vdots b)$ with $-M_{.1}$ as the pivot column and row 1 as the pivot row. Suppose this leads to

$$\begin{pmatrix} -m_{11} & -m_{12} & \cdots & -m_{1m} & \vdots & b_1 \\ 0 & -\tilde{m}_{22} & \cdots & -\tilde{m}_{2m} & \vdots & \tilde{b}_2 \\ \vdots & & & \vdots & & \\ 0 & -\tilde{m}_{m2} & \cdots & -\tilde{m}_{mm} & \vdots & \tilde{b}_m \end{pmatrix}$$

Let $\tilde{\mathcal{M}} = (\tilde{m}_{ij} : i, j = 2 \text{ to } m)$, $\tilde{b} = (\tilde{b}_2, \cdots, \tilde{b}_m)^T$, $\omega = (\mu_2, \cdots, \mu_m)$, $\xi = (\lambda_2, \cdots, \lambda_m)^T$. Then the lower order LCP obtained after eliminating λ_1, from (6) is

$$\omega - \tilde{\mathcal{M}}\xi = \tilde{b} \ , \quad \omega, \ \xi \geq 0 \ . \quad \omega^T \xi = 0 \ , \tag{7}$$

Since 1 is a critical index, every solution for (3) comes from a solution to (7), with $\mu_1 = 0$ and λ_1 obtained from the first equation in (6).

Since M is PSD, symmetric and of rank n, from the manner in which $\tilde{\mathcal{M}}$ is obtained, $\tilde{\mathcal{M}}$ of order $(m-1) \times (m-1)$ is PSD, symmetric, and of rank $(n-1)$. Also, since b is in the linear hull of the columns of

M, $\tilde{b}$ is in the linear hull of the columns of $\check{\mathcal{M}}$ So $\check{\mathcal{M}}$ has a Cholesky factor, $\tilde{\mathcal{Q}}$ of order $(n-1) \times (m-1)$ and rank $(n-1)$ and $\tilde{b}$ is in the linear hull of the columns of $\tilde{\mathcal{Q}}$, i.e., the system, $\tilde{\mathcal{Q}}^T\tilde{y} = -\tilde{b}$, has a solution $\tilde{y} \in \mathbf{R^{n-1}}$. Thus (7) is equivalent to the lower order NPP $[\tilde{\mathcal{Q}}; \tilde{y}]$ where $\tilde{\mathcal{Q}}$ is of order $(n-1) \times (m-1)$ and rank $n-1$.

Hence, if a critical index for (1) is known, it can be reduced to a lower dimensional problem

Reduction Using Geometrical Arguments

Given a critical index for the NPP $[Q; q]$, its reduction to a lower dimensional NPP can also be carried out using geometric arguments as discussed[7]. Again, suppose 1 is a critical index. Then the nearest point to q in Pos(Q) is also the nearest point to q in Pos($(Q \vdots -Q_{.1})$). Define

$$Q^-_{.j} = Q_{.j} - \frac{Q_{.1}(Q_{.1})^T Q_{.j}}{\|Q_{.1}\|^2} \ , \quad j = 2 \text{ to } n$$

$$q^- = q - \frac{Q_{.1}(Q_{.1})^T q}{\|Q_{.1}\|^2} \ .$$

For $j = 2$ to n, $Q^-_{.j}$ is the orthogonal projection of $Q_{.j}$ in the hyperplane through the origin orthogonal to the ray of $Q_{.1}$, likewise q^- is the orthogonal projection of q in the same hyperplane. The cone Pos($(Q \vdots -Q_{.1})$) is the direct sum of the full line generated by $Q_{.1}$ and the cone Pos(Q). Let $Q^- = (Q^-_{.2} \cdots Q^-_{.m})$. If x^{-*} is the solution of the NPP $[Q^-; q^-]$ as embedded in $\mathbf{R^n}$, then $x^* = x^{-*} + \frac{Q_{.1}(Q_{.1})^T q}{\|Q_{.1}\|^2}$ is the solution of the original NPP $[Q; q]$.

Q^- is of order $n \times (m-1)$, and from the construction it is clear that the rank of Q^- is one less than the rank of Q, hence the set of row vectors of Q^- form a linearly dependent set. It can be verified that $(Q^-)^T(Q^-) = \tilde{\mathcal{Q}}$ discussed earlier, and that the NPP $[Q^-; q^-]$ is equivalent to the LCP $(\tilde{b}, \check{\mathcal{M}})$. Since Q^- is not of full row rank, it is possible to transform the NPP $[Q^-; q^-]$ into an equivalent NPP of the form $[Q^=; q^=]$, where $Q^=$ is obtained from Q^- by dropping a dependent row vector from it, as discussed in Section 1, but it is not necessary to do this work.

4. CONDITIONS FOR CRITICALITY

Given $z \in R^n$, $z \neq q$, define $\mathbf{B(q; z)} = \{\mathbf{x} : \|\mathbf{x} - \mathbf{q}\| < \|\mathbf{x} - \mathbf{z}\|\} \subset \mathbf{R^n}$. $\mathbf{T(q; z)} = \{\mathbf{x} : (\mathbf{x} - \mathbf{z})^\mathbf{T}(\mathbf{q} - \mathbf{z}) = \mathbf{0}\}$ is the tangent plane to $\mathbf{B(q; z)}$ at its boundary point z.

Property 1: The point $z \in \mathbf{R^n}$ is said to satisfy property 1 if $z \neq 0$ and it is the nearest point (by Euclidean distance) to q on its ray $\{\gamma z : \gamma \geq 0\}$.

It can be verified that z satisfies property 1 iff $z \neq 0$ and $z^T(q - z) = 0$, i.e., $0 \in \mathbf{T}(q; z)$. For such points, we therefore have $\mathbf{T}(q; z) = \{x : x^T(q - z) = 0\}$.

The open half space defined by $\mathbf{T}(q; z)$ containing q is called the *near side* of $\mathbf{T}(q; z)$, while its complement is called the *far side* of $\mathbf{T}(q; z)$. So, for points z satisfying property 1, near side of $\mathbf{T(q; z)}$ is $\{x : x^T(q - z) > 0\}$, and the far side of $\mathbf{T(q; z)}$ is $\{x : x^T(q - z) \leq 0\}$

For z satisfying property 1, we define

$$\mathbf{N}(z) = \{j : j = 1 \text{ to } m \text{ and } (Q_{.j})^T(q - z) > 0\} ,$$

it is the set of j such that $Q_{.j}$ is on the near side of $\mathbf{T}(q; z)$.

Let $\mathbf{\Gamma} = \{1, \cdots, m\}$. For $\phi \neq \mathbf{S} \subset \mathbf{\Gamma}$, we define

$Q_{.\mathbf{S}} =$ matrix of order $n \times |\mathbf{S}|$ whose column vectors are $Q_{.j}$ for $j \in \mathbf{S}$.

$\mathbf{H(S)} =$ linear hull of the column vectors of $Q_{.\mathbf{S}}$ in $\mathbf{R^n}$.

$q(\mathbf{S}) =$ orthogonal projection of q in $\mathbf{H(S)}$.

In the algorithm, we will deal with subsets $\mathbf{S} \subset \mathbf{\Gamma}$ satisfying: $\{ Q_{.j} : j \in \mathbf{S} \}$ is linearly independent. For such sets $\mathbf{S}$, we have $q(\mathbf{S}) = Q_{.\mathbf{S}}((Q_{.\mathbf{S}})^T Q_{.\mathbf{S}})^{-1}(Q_{.\mathbf{S}})^T q$.

We have the following Theorems 2, 3 based on corresponding results[7] for the NPP in simplicial cones.

Theorem 2: A point $\bar{x} \in \text{Pos}(Q)$ is the nearest point in $\text{Pos}(Q)$ to q iff

(i) $\bar{x}$ satisfies property 1, and

(ii) either $\bar{x} = q$ itself, or $\mathbf{N}(\bar{x}) = \phi$.

Proof: If $\bar{x} = q$, $q \in \text{Pos}(Q)$ and hence q is the nearest point in $\text{Pos}(Q)$ to itself. So, assume that $\bar{x} \neq q$.

Since $\bar{x} \in \text{Pos}(Q)$, it is the nearest point in $\text{Pos}(Q)$ to q iff $\text{Pos}(Q) \cap \mathbf{B}(q; \bar{x}) = \phi$, i.e., iff $\bar{x}$ is a boundary point of $\text{Pos}(Q)$, and $\mathbf{T}(q; \bar{x})$ separates $\text{Pos}(Q)$ and $\mathbf{B}(q; \bar{x})$ which happens iff $\mathbf{N}(\bar{x}) = \phi$. ∎

Theorem 3: Assume that $q \notin \text{Pos}(Q)$. Let $\bar{x} \in \text{Pos}(Q)$ satisfy property 1. If $\mathbf{N}(\bar{x})$ is a singleton set, $\{h\}$, then h is a critical index for the NPP $[Q; q]$.

Proof: Since $\mathbf{N}(\bar{x}) \neq \phi$, by Theorem 2, $\bar{x}$ is not the nearest point in Pos(Q) to q. Let x^* be the nearest point in Pos(Q) to q. So $\|x^* - q\| < \|\bar{x} - q\|$ and hence $x^* \in \mathbf{B}(q; \bar{x})$.

Since $\bar{x}$ satisfies property 1, $0 \in \mathbf{T}(q; \bar{x})$. Also, since $\mathbf{N}(\bar{x}) = \{h\}$, $Q_{.j}$ is on the far side of $\mathbf{T}(q; \bar{x})$, that is $\mathbf{T}(q; \bar{x})$ separates the ray of $Q_{.j}$ from $\mathbf{B}(q; \bar{x})$, for all $j \neq h$. Hence $\mathbf{T}(q; \bar{x})$ separates Pos$(Q_{.1}, \cdots, Q_{.h-1}, Q_{.h+1}, \cdots Q_{.m})$ and $\mathbf{B}(q; \bar{x})$.

These facts imply that Pos(Q) $\cap$ $\mathbf{B}(q; \bar{x}) \neq \phi$, and that none of the points in it is in Pos$(Q_{.1}, \cdots, Q_{.h-1}, Q_{.h+1}, \cdots, Q_{.m})$. Hence, x^* satisfies this property too, that is, if $x^* = \sum_{j=1}^{m} \gamma_j Q_{.j}$ with $\gamma_j \geq 0$ for all j, then we must have $\gamma_h > 0$. Therefore h is a critical index for the NPP $[Q; q]$. ∎

5. THE ALGORITHM

If $(Q_{.j})^T q \leq 0$ for all $j = 1$ to m, $\mathbf{N}(0) = o$, and hence 0 is the nearest point in Pos(Q) to q by Theorem 2, in this case $(\mu^* = -Q^T q, \lambda^* = 0)$ is the solution of the LCP (3).

So, in the sequel we assume that $(Q_{.j})^T q > 0$ for at least one j. The algorithm applies a routine for finding a critical index at most $n - 2$ times. This routine itself may either terminate with the nearest point (in which case the whole algorithm terminates), or with a critical index. In the latter case we use the critical index to reduce the NPP into one of lower dimension as discussed above, and repeat the whole procedure on the reduced problem. As the rank reduces by 1 with each reduction, this routine will be called at most $(n - 1)$ times during the algorithm. Actually, when the rank becomes 2, the reduced problem can be solved by the special direct algorithm discussed in Section 2, and then from this solution build up an optimum combination vector for the original problem using equations of the form of the first in (6) from earlier reduction Steps.

The routine maintains a subset $\mathbf{S} \subset \Gamma$ such that $\{Q_{.j} : j \in \mathbf{S}\}$ is linearly independent, and a current point $x \in$ Pos$(Q_{.\mathbf{S}})$ which always satisfies Property 1. $\bar{x}$ gets closer to q as the algorithm progresses. $\bar{\lambda} = (\bar{\lambda}_j) \geq 0$ is the combination vector corresponding to $\bar{x}$, satisfying: $\bar{\lambda}_j = 0$ for all $j \notin \mathbf{S}$, and $Q\bar{\lambda} = \bar{x}$

Routine for Finding a Critical Index

Step 1: [Initialization]: For each $j = 1$ to m, define q^j as in (2). q^j is the nearest point to q in the ray of $Q_{.j}$. If $q^j = 0$ for all $j = 1$ to m, 0 is the nearest point in Pos(Q) to q, terminate the whole algorithm.

If $q^j \neq 0$ for at least one j, let q^h be the nearest to q among all q^j, break ties arbitrarily.

Set $\bar{x} = q^h$, it satisfies property 1. Let $\mathbf{S} = \{h\}$. Define the corresponding $\bar{\lambda} = \bar{\lambda}_j$ where $\bar{\lambda}_h = \frac{((Q_{.h})^T q)}{\|Q_{.h}\|^2}$, and $\bar{\lambda}_j = 0$ for $j \neq h$.

Step 2: Compute $\mathbf{N}(\bar{x}) = \{j : 1 \leq j \leq m \text{ and } (Q_{.j})^T(q - \bar{x}) > 0\}$. If $\mathbf{N}(\bar{x}) = \phi$, $\bar{x}$ is the nearest point in Pos(Q) to q, and the corresponding $\bar{\lambda}$ is an optimum combination vector, terminate the whole algorithm.

If $\mathbf{N}(\bar{x})$ is a singleton set $\{h\}$, h is a critical index for the present nearest point problem, terminate this routine.

If $|\mathbf{N}(\bar{x})| \geq 2$, go to Step 3.

Step 3: If $\mathbf{N}(\bar{x}) \cap (\mathbf{\Gamma} \backslash \mathbf{S}) = \phi$ go to Step 5, otherwise go to Step 4.

Step 4: Select a $\rho \in \mathbf{N}(\bar{x}) \cap (\mathbf{\Gamma} \backslash \mathbf{S})$. Compute $\hat{q}$, the orthogonal projection of q onto the linear hull of $\{\bar{x}, Q_{.\rho}\}$. $\hat{q}$ will be in Pos($\bar{x}, Q_{.\rho}$). Actually $\hat{q} = \alpha_1 \bar{x} + \alpha_2 Q_{.\rho}$ where both $\alpha_1 > 0$ and $\alpha_2 > 0$. Define $\hat{\lambda} = (\hat{\lambda}_j)$ where $\hat{\lambda}_j = \alpha_1 \bar{\lambda}_j$ for $j \in \mathbf{S}$, α_2 for $j = P$, and 0 otherwise. Let $\mathbf{S_1} = \mathbf{S} \cup \{\rho\}$. Obtain $(Q^T_{.\mathbf{S}_1} Q_{.\mathbf{S}_1})^{-1}$ from $(Q^T_{.\mathbf{S}} Q_{.\mathbf{S}})^{-1}$ using any of the updating schemes. If linear dependence of $Q_{.\rho}$ with columns of $Q_{.\mathbf{S}}$ is signaled, go to Step 5, otherwise replace $\mathbf{S}$, $\bar{x}$, $\bar{\lambda}$ by $\mathbf{S_1}$, $\hat{q}$, $\hat{\lambda}$ respectively and go back to Step 2.

Step 5 Compute $q(\mathbf{S}) = Q_{.\mathbf{S}} \alpha(\mathbf{S})$, where $\alpha(\mathbf{S}) = (\alpha_j(\mathbf{S}) : j \in \mathbf{S}) = ((Q_{.\mathbf{S}})^T Q_{.\mathbf{S}})^{-1} (Q_{.\mathbf{S}})^T q$.

If $\alpha(\mathbf{S}) \geq 0$, define $\hat{\lambda} = (\hat{\lambda}_j)$ by $\hat{\lambda}_j = \alpha_j(\mathbf{S})$ if $j \in \mathbf{S}$, 0 otherwise. Replace $\bar{x}$, $\bar{\lambda}$ by $q(\mathbf{S})$, $\hat{\lambda}$ respectively and go back to Step 2.

If $\alpha(\mathbf{S}) \not\geq 0$, go to Step 6.

Step 6 Let $\hat{x}$ denote the last point in Pos(Q) as we move from $\bar{x}$ along the line segment joining it to $q(\mathbf{S})$. Actually

$$\hat{x} = \sum_{j \in \mathbf{S}} \left((1-\beta)\bar{\lambda}_j + \beta\alpha_j(\mathbf{S}) \right) Q_{.j}$$

where

$$\beta = \min \left\{ \frac{\bar{\lambda}_j}{\bar{\lambda}_j - \alpha_j(\mathbf{S})} : j \in \mathbf{S} \text{ such that } \alpha_j(\mathbf{S}) < 0 \right\} \qquad (8)$$

Let $k \in \mathbf{S}$ attains the minimum in (8), break ties arbitrarily. Define $\hat{\lambda} = (\hat{\lambda}_j)$ where $\hat{\lambda}_j = (1-\beta)\bar{\lambda}_j + \beta\alpha_j(\mathbf{S})$, and 0 for $j \notin \mathbf{S}$. Replace $\bar{x}$, $\bar{\lambda}$, by $\hat{x}$, $\hat{\lambda}$, respectively, delete k from $\mathbf{S}$, and go to Step 5.

Finite Convergence of the Algorithm

We have the following facts:

1. At each pass through Step 4, $\|\bar{x} - q\|$ strictly decreases. At each pass through Steps 5 and 6, $\|\bar{x} - q\|$ may decrease or stay the same.

2. Step 4 can be performed at most n times consecutively before linear dependence is signaled.

3. Steps 5 and 6 could be performed at most $n-1$ times consectively before the projection gets into the relative interior of the Pos of the current set.

4. The same subset of $\mathbf{\Gamma}$ cannot reappear as S once it changes due to 1.

Since there are only a finite number of subsets of $\mathbf{\Gamma}$, these facts imply that the routine must terminate in a finite number of steps. Also, the overall algorithm is finite as it uses the above routine at most $(n-2)$ times.

6. IMPLEMENTATION AND RELATED NUMERICAL ISSUES

$((Q_{.\mathbf{S}})^T Q_{.\mathbf{S}})^{-1}$ can be updated by standard updating schemes for projection matrices whenever $\mathbf{S}$ changes (as it always changes by the addition or deletion of one element). These schemes will detect linear dependence of the set of column vectors of the new $Q_{.\mathbf{S}}$, if it occurs when a new element is added to the set $\mathbf{S}$. An even better implementation is obtained by updating and using the Cholesky factor of $(Q_{.\mathbf{S}})^T Q_{.\mathbf{S}}$ instead of its inverse, this cuts down the amount of work for each updating from $O(|\mathbf{S}|^3)$ to $O(|\mathbf{S}|^2)$.

Computing $\mathbf{N}(\bar{x})$ completely in Step 2 is expensive, particularly when m is large. We found that an efficient way to carry out Step 2 and to select a ρ for carrying out Step 4 if that is the step to move next, is to begin the search for the new ρ from the previous ρ, continue upto m, and then from 1 upto the previous ρ again, until the first eligible candidate is noticed, at which point the search is terminated by selecting that candidate as the new ρ. This is similar to the LRC (least recently considered) entering index strategy commonly mentioned in the literature on the Simplex algorithm for linear programming.

Computational Experience

The above algorithm was coded in FORTRAN 77 and tested using APOLLO series 4000 machines. The performance of the algorithm is compared with the performance of the following two algorithms. K. Haskell and R. Hanson[4] for solving linearly constrained least square problems (HH) which - in our case - boils down to their implementation of Lawson and Hanson[5] for Pos cones. The code is available as ACM software 587. The other algorithm WIL is that of D.R. Wilhelmsen[8]. We coded Wilhelmsen's algorithm using FORTRAN 77. We used updating procedures based on the Cholesky factor. And we used LRC (least recent considered) entering rule as discussed above. Surprisingly this happens to be far superior to other rules. These strategies accelerate this algorithm a lot.

Test problems were constructed by generating the elements of Q to be uniformly distributed between -5 and 5 and the elements of q to be uniformly between -20 and 20. All the test problems are fully dense.

The timings of all three algorithms are shown in Table 1. It is clear from the table that our algorithm is 2-4 times faster than the

Table 1: Summary of Performance Results
(Average time per problem in APOLLO 4000 seconds
on fully dense randomly generated problems.)

n	m	# problems	our algorithm	HH	WIL
50	70	10	6.84	8.53	9.55
150	150	10	49.9	74.1	93.6
200	250	10	180.3	412.5	400.1
300	400	10	790.0	1544.0	1644.1
400	500	5	1194.5	3038.9	3118.4
500	550	5	1470.4	4778.5	4364.4
600	800	3	5285.0	11927.0	13260.7

Table 2: Average number of type (i) and (ii) projections per problem.

n	m	# type (i) projections	# type (ii) projections
50	70	52.8	3.5
100	150	116.4	4.5
200	250	177.6	3.7
300	400	303.4	4.2
400	500	351.6	3.9
500	550	357.2	3.4
600	800	587.0	4.67

HH and the Wilhelmsen algorithms on an average. Moreover, the algorithm performs much better than HH when the cone is close to simplicial (or when m is comparable to n) while marginally better when $m \gg n$. This may be due to the fact that computationally cheap two ray series of projections (Step 4 of the algorithm) becomes less effective when $m \gg n$.

The work in our algorithm consists of three types of steps, (i) the orthogonal projection of a point into a two dimensional subspace, (ii) orthogonal projection of a point into a subspace of dimension > 2, and (iii) reduction of the problem into one of lower dimension using a critical index. Among steps (i) and (ii), clearly (ii) is far more expensive than (i). In Table 2, we provide the total number of steps of types (i) and (ii) carried out on an average per problem in our algorithm as the order of Q ranges from 50×70 to 600×800. The number of steps of type (i) clearly grows as the order of Q increases,

but the number of steps of type (ii) seems to be more or less constant (between 3 to 5). The good performance of our algorithm stems from the fact that the majority of work in it involves steps of type (i) which are computationally cheap.

ACKNOWLEDGEMENTS

This work was partially supported by NSF Grant No. ECS-8521183 and by King Fahd University of Petroleum and Minerals.

7. REFERENCES

1. Chang, S.Y. and Murty, K.G. (1989) The Steepest Descent Gravitational Method for Linear Programming, *Discrete Applied Mathematics*, 25, 211-239.

2. Gilbert, E.G., Johnson, D.W., and Keerthi, S.S. (1988) A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space. *IEEE Journal of Robotics and Automation*, 4 (2), 193-203.

3. Golub, G.H.and Van Loan, C.F. (1989) *Matrix Computations*, Baltimore, MD, John Hopkins University Press.

4. Haskell, K. and Hanson, R. (1981) An Algorithm for Linear Least Squares Problems with Equality and Nonnegativity Constraints, *Mathematical Programming*, 21, 98-118.

5. Lawson, C.L. and Hanson, R.J. (1974) *Solving Least Squares Problems*, Englewood Cliffs, NJ, Prentice-Hall, Inc.

6. Murty, K.G. (1988) *Linear Complementarity, Linear and Nonlinear Programming*, West Berlin, Heldermann Verlag.

7. Murty, K.G. and Fathi, Y. (1982) A Critical Index Algorithm for the Nearest Point Problem on Simplicial Cones, *Mathematical Programming*, 23, 206-215.

8. Wilhelmsen, D.R. (1976) A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear Approximations, *Mathematics of Computation*, 30, 48-57.

9. Wolfe, P. (1974) Algorithm for a Least Distance Programming, *Mathematical Programming Study I*, 190-205.

10. Wolfe, P. (1976) Finding Nearest Point in a Polytope, *Mathematical Programming*, 11, 128-149.