1971

A Simplex Algorithm - Gradient Projection Method for Nonlinear Programming

L. Duane Pyle

Report Number:
71-055


This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
CSD TR 55

"A SIMPLEX ALGORITHM - GRADIENT PROJECTION METHOD FOR NONLINEAR PROGRAMMING".

by

L. Duane Pyle

10 June 1971

ABSTRACT

Witzgall [7], commenting on the gradient projection methods of R. Frisch and J. B. Rosen, states: "More or less all algorithms for solving the linear programming problem are known to be modifications of an algorithm for matrix inversion. Thus the simplex method corresponds to the Gauss-Jordan method. The methods of Frisch and Rosen are based on an interesting method for inverting symmetric matrices. However, this method is not a happy one, considered from the numerical point of view, and this seems to account for the relative instability of the projection methods".

This paper presents an implementation of the gradient projection method which uses a variation of the simplex algorithm.

The underlying (well-known) geometric idea is that the simplex algorithm for linear programming [1] provides a method for obtaining vectors along the "edges" [4] of the feasible region A={x|Ax=b, x≥0} which lie in certain null spaces. This property is discussed in
detail in section §1., Geometric Analysis of the Simplex Method of Linear Programming. In section §2., Projection on Faces of A of Higher Dimension, the geometric analysis of §1. is extended to obtain the orthogonal projection matrix P such that

$$ R(P) = N(A) \bigcap_{i=6+1} M(u(1)) $$

where $R(P)$ is the range of $P$; $N(A)$ is the null space of $A$; and $M(u(1)) = \{ x | x_i = 0 \}$.

The gradient projection method [6], [2] requires computations involving (1) an orthogonal projection matrix whose range is a certain null space; and (2) a related generalized inverse [3]. In section §3., Simplex Algorithm Implementation of the Gradient Projection Method, the developments given in §2. are combined with the simplex algorithm to provide the computational results required by the gradient projection method. Motivation for this approach may be found in [5].

In the approach given here, a representation of the projection matrix

$$ P = (I-NN^*) $$

is generated using the simplex algorithm, whereas Rosen gives a method for obtaining $NN^*$ based on an algorithm involving $(N^TN)^{-1}$. ($N$ is a matrix whose columns are normals to the "active" constraints.) If the dimension of $R(I-NN^*)$ is small compared to the dimension of $R(NN^*)$, as is the case when the current vector
iterate lies on a face of \( \Lambda \) of low dimension, one would expect significant computational improvements. This expectation is further enhanced by the use of a variation of the product form of the inverse in computing the vectors which constitute the representation of the matrix \( P \), and by the use of "simplex multipliers" and "relative cost factors" in the standard fashion of simplex algorithm technology.
§1. Geometric Analysis of the Simplex Algorithm of Linear Programming.

The notation used here is similar to that in [1], Chapters 5 and 8.

Consider the linear programming problem

Minimize \((x, c)\) where

\[
(1.01) \quad A x = \begin{bmatrix} P_1 & P_2 & \cdots & P_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}
\]

and \(x \geq 0\), where 0 is a column vector of zeros. Reformulate this problem as

Minimize \(
\begin{bmatrix} z \\ x_1 \\ \vdots \\ x_n \end{bmatrix}
\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\)

\(= z\) where

\[
(1.02) \quad \begin{bmatrix} P_1 & P_2 & \cdots & P_n \end{bmatrix} \begin{bmatrix} z \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} -1 & c_1 & \cdots & c_n \\ 0 & P_1 & \cdots & P_n \end{bmatrix} \begin{bmatrix} z \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ P_0 \end{bmatrix} = \begin{bmatrix} 0 \\ x_0 \end{bmatrix}
\]

\(x_1 \geq 0 \ (i = 1, \ldots, n)\),

\(z\) unrestricted.
Now, suppose $x_1, x_2, \ldots, x_m$ and $z$ are basic, feasible
variables (we assume that $A$ is $m$ by $n$ and that rank $A = m$; we
further assume that index sets have convenient labels,
such as $(1, 2, \ldots, m)$, rather than using the correct, but
ponderous $(j_1, j_2, \ldots, j_m)$).

(1.03) With $\hat{B} = [\hat{P}_2 \hat{P}_1 \ldots \hat{P}_m]$, where $\hat{P}_2 = [\hat{I}]$, after a sequence of
one pivotal reductions obtain the canonical form

(1.04) $\hat{B}^{-1}[\hat{P}_2 \hat{P}_1 \ldots \hat{P}_m \hat{P}_{m+1} \ldots \hat{P}_n \hat{P}_0] = [\hat{I} \hat{P}_{m+1} \ldots \hat{P}_n \hat{P}_0]$

which we partition as

(1.05) $\begin{bmatrix}
-1 & \theta^T & \bar{c}_{m+1} \ldots \bar{c}_n & \bar{z}_0 \\
\theta & I & \hat{P}_{m+1} \ldots \hat{P}_n & \hat{P}_0
\end{bmatrix}$

(1.06) From $\hat{P}_j = \hat{B}^{-1} \hat{P}_j$ for $(j = 0, m+1, \ldots, n)$.

(1.07) one obtains $\hat{P}_j = \hat{B} \hat{P}_j$, thus

(1.08) $\hat{P}_j = [P_1 \ldots P_m] [P_j] (i = 0, m+1, \ldots, n)$

(1.09) $\bar{c}_j = [c_1 \ldots c_m] [P_j] + \bar{c}_j (j = m+1, \ldots, n)$

and

(1.10) $\bar{c}_0 = [c_1 \ldots c_m] [P_0] - \bar{z}_0$.

(1.08) states that

(1.11) $[P_1 P_2 \ldots P_m P_{m+1} \ldots P_j \ldots P_n] \begin{bmatrix}
P_j \\
\theta_1 \\
\theta_2
\end{bmatrix} = 0$
where $\theta_1$ and $\theta_2$ are vectors of zeros; the scalar -1 is the $j$th component of the vector
\[
\begin{bmatrix}
\alpha_j \\
\theta_1 \\
-1 \\
\theta_2
\end{bmatrix}
\]
which thus lies in the null space of the matrix $A$.

If the basic solution corresponding to the canonical form (1.04) is non-degenerate, that is, if $P_0 > 0$, then the point
\[
\begin{bmatrix}
x_1 \\
\vdots \\
x_m \\
x_{m+1} \\
\vdots \\
x_j \\
\vdots \\
x_n
\end{bmatrix}
= 
\begin{bmatrix}
P_0 \\
\theta_1 \\
1 \\
\theta_2
\end{bmatrix}
+ \alpha
\]
is feasible for problem (1.01) provided $\alpha > 0$, $\alpha$ sufficiently small.

The vector
\[
\begin{bmatrix}
-\alpha_j \\
\theta_1 \\
1 \\
\theta_2
\end{bmatrix}
\]
is a vector along an edge of
\[
A = \{x \mid Ax = b, x \geq 0\}^* \text{ [5].}
\]
The question is whether or
not a path in the direction of this vector produces a
decrease in \((x,c)\). This will be the case for \(\alpha > 0\),
provided that

\[
(1.14) \left( \begin{array}{c}
-\frac{\partial f}{\partial x_1} \\
\frac{\partial f}{\partial x_2} \\
1 \\
\end{array} \right), c < 0. \text{ Now, from relation (1.09)}
\]

we have that \(\left( \begin{array}{c}
-\frac{\partial f}{\partial x_1} \\
\frac{\partial f}{\partial x_2} \\
1 \\
\end{array} \right), c = \overline{c}_j\), thus if \(\overline{c}_j < 0\) a decrease

in \((x,c)\) results for all \(\alpha > 0\).

§2. Projection on Faces of \(\Lambda\) of Higher Dimension.

From the analysis given in §1, it is apparent that a
non-degenerate, basic feasible canonical form provides,
explainingly, the directions of the projections of the
gradient of the function \(f(x) = (x,c)\), denoted

\[
\nabla f = \left[ \begin{array}{c}
\frac{\partial f}{\partial x_1} \\
\vdots \\
\frac{\partial f}{\partial x_n}
\end{array} \right]
\]

when \(\nabla f\) is projected orthogonally upon
the various 1-dimensional faces of \( \Lambda = \{ x | Ax = b, x \geq \theta \} \) which intersect in the vertex of \( \Lambda \) associated with that canonical form. In order to obtain the projections of \( \nu_f \) upon higher dimensional faces of \( \Lambda \), consider the following:

Suppose the canonical form

\[
\begin{bmatrix}
1 & \mathbb{F}_{m+1} \cdots \mathbb{F}_n
\end{bmatrix}
\]

(2.01) has been obtained.

(Note the omission of the "z-row" and \( \mathbb{F}_0 \) column.) Define

\[
\mathbb{F}^{(m+1)} = \begin{bmatrix}
-\mathbb{F}^{m+1} \\
1 \\
\theta
\end{bmatrix}
\quad \text{and} \quad
\mathbf{e}^{(m+1)} = \frac{1}{\| \mathbb{F}^{(m+1)} \|} \mathbf{e}^{(m+1)}.
\]

(2.02)

Then from \( \S 1 \) it follows that

\[
\mathbf{e}^{(m+1)} \in \eta(A) \cap \mathbb{M}^{(m+2)} \cap \cdots \cap \mathbb{M}^{(n)},
\]

where

\[
\eta(A) = \text{the null space of } A \\
= \{ z | Az = 0 \}.
\]

Now, adjoin \( \eta^{(m+1)} \) to (2.01) and complete the reduction to "canonical form"; i.e. using elementary row operations in the obvious way, reduce the form
\[
\begin{equation}
(2.03) \begin{bmatrix}
I & F_{m+1} & \cdots & F_n \\
-\frac{F^T}{m+1} & 1 & \theta^T
\end{bmatrix}
\end{equation}
to the form

\[
(2.04) \begin{bmatrix}
I & F_{m+2} & \cdots & F_n \\
-\frac{F^T}{m+2} & 1 & \theta
\end{bmatrix}
\]
where \( I \) is \((m+1)\) by \((m+1)\).

\[
(2.05) \quad \text{Define } \eta^{(m+2)} = \begin{bmatrix}
-\frac{F}{m+2} \\
1 \\
\theta
\end{bmatrix}
\text{ and } \eta^{(m+2)} = \frac{1}{\|\eta^{(m+2)}\|} \eta^{(m+2)}.
\]

As before, it follows that

\[
(2.06) \quad \eta^{(m+2)} \in \mathcal{H}(A) \cap \mathcal{H}(\eta^{(m+1)})^\perp \cap \mathcal{H}(u^{(m+3)})^\perp \cap \cdots \cap \mathcal{H}(u^{(n)})^\perp.
\]

Continuing in this manner, obtain vectors

\[
(2.07) \quad \eta^{(m+1)}, \eta^{(m+2)}, \ldots, \eta^{(s)} \text{ where } (\eta^{(i)}, \eta^{(j)}) = \delta_{ij}
\]

\[(s = m+1, m+2, \ldots, n)\]

and

\[
(2.08) \quad \eta^{(i)} \in \mathcal{H}(A) \cap \mathcal{H}(u^{(s+1)})^\perp \cap \cdots \cap \mathcal{H}(u^{(n)})^\perp
\]

for \((i = (m+1),(m+2), \ldots, s)\) where \( s \leq n \).

Thus, as is well-known,

\[
(2.09) \quad P = \sum_{i=m+1}^s \eta^{(i)} \eta^{(i)T}
\]
is an orthogonal projection and we have
Theorem 2.1:

(2.10) \[ \mathcal{R}(P) = \eta(A) \cap \bigwedge_{i=s+1}^{n} \mathcal{m}(u(i)) \perp, \text{ where } \mathcal{R}(P) = \text{Range of } \]
\[ P = \{ y \in \mathbb{F} \mid z \text{ with } y = Pz \}. \]

Proof:

Relation (2.10) follows from the assumption that the vertex associated with the canonical form (2.01) is non-degenerate (see definitions of degeneracy and non-degeneracy given in [4]), for then the vectors
\[ u(m+1), ..., u(n), \]
together with a basis for the column space of \( A^T \), \( \{ e(1) \} (i = q+1, ..., n) \) where \( q = n-m \), form a set of \( n \), linearly independent vectors. Under the assumption that \( A \) has full row rank, one could just as well take the \( \{ e(1) \} \) to consist of the columns of \( A^T \).

The origin of the notation \( \{ e(1) \} \) resides in developments given in [4] and [5]. Thus, by construction

\[ \eta(m+1) e \in \mathcal{m}(e(q+1), ..., e(n), u(m+2), ..., u(n)) \perp \]
\[ = \eta(e(q+1), ..., e(n)) \perp \bigwedge_{i=m+2}^{n} \mathcal{m}(u(i)) \perp \]
\[ = \eta(A) \cap \bigwedge_{i=m+2}^{n} \mathcal{m}(u(i)) \perp \]

(2.11) where dimension of \( \mathcal{m}(e(q+1), ..., e(n), u(m+2), ..., e(n)) \perp = 1. \)

But dimension of \( \mathcal{R}(\eta(m+1) \eta(m+1)^T) = 1 \), together with
\[ \mathcal{R}(\eta^{(m+1)}\eta^{(m+1)T}) \subset \eta(A) \bigcap_{i=m+2}^{n} \mathfrak{M}(u^{(1)})^\perp \]

implies

\[ \mathcal{R}(\eta^{(m+1)}\eta^{(m+1)T}) = \eta(A) \bigcap_{i=m+2}^{n} \mathfrak{M}(u^{(1)})^\perp . \]

Similarly,

\[ \eta^{(m+2)} \in \eta(A) \bigcap_{i=m+3}^{n} \mathfrak{M}(u^{(1)})^\perp \]

where dimension of \( \eta(A) \bigcap_{i=m+3}^{n} \mathfrak{M}(u^{(1)})^\perp = 2 \),

and so on. Thus

\[ \mathcal{R}(P) \subset \eta(A) \bigcap_{i=s+1}^{n} \mathfrak{M}(u^{(1)})^\perp \]

where \( \text{dim } \mathcal{R}(P) = (s+1) - (m+1) = s - m \) and

\[ \text{dim } \eta(A) \bigcap_{i=s+1}^{n} \mathfrak{M}(u^{(1)})^\perp = n - [(n-q) + (n-s)] = s - m, \]

consequently

\[ \mathcal{R}(P) = \eta(A) \bigcap_{i=s+1}^{n} \mathfrak{M}(u^{(1)})^\perp . \]

Remark:

When the set of unit vectors is exhausted, i.e.,
when \( s = n \), then \( \mathcal{R}(P) = \eta(A) \) and

\[ P = I - A^+ A, \]

where \( A^+ \) is the generalized inverse of \( A \) [3].

In §1 the Simplex Algorithm of linear programming [1] was shown to provide a method for obtaining "vectors along the edges" [5] of the feasible region, \( A \), which lie in certain null spaces. This idea was extended in §2 to provide a means of projecting on faces of higher dimension. Use of the gradient projection method [6] requires computations involving

(1) an orthogonal projection matrix whose range is a certain null space; and (2) a related generalized inverse [6],[2]. In this section we continue the discussion given in §2 to show how the simplex algorithm may be used to provide the computational results involving (1) and (2). Motivation for this approach may be found in [4].

Consider the transformation which must be applied to the matrix discussed in §2:

\[
\begin{bmatrix}
I \\
\vdots \\
\mathbf{P}_{m+1} \\
\vdots \\
\mathbf{P}_n
\end{bmatrix}
\begin{bmatrix}
\mathbf{P}^T \\
1 \\
\mathbf{g}^T
\end{bmatrix}
\]

Let

\[
\begin{bmatrix}
\alpha_{1,m+1} \\
\vdots \\
\alpha_{m,m+1}
\end{bmatrix}
\]

(3.02)

and

\[
\begin{bmatrix}
\mathbf{P}_{m+1} \\
\vdots \\
\mathbf{P}_{m+1}
\end{bmatrix}
\]
(3.03) \[ m_{m+1} = 1 + \frac{1}{\sigma} \sum_{i=1}^{m} \alpha_{i,m+1} = 1 + (\mathbf{F}_{m+1}, \mathbf{F}_{m+1}). \]

If \( m_{m+1} = 1 \) then no action is required. Otherwise \( m_{m+1} > 1 \) and the transformation of (3.01) to canonical form

(3.04) \[
\begin{bmatrix}
1 & \mathbf{F}_{m+2} & \ldots & \mathbf{F}_{n}
\end{bmatrix}
\]

is well-defined. This is achieved by pre-multiplying the columns of (3.01) by

(3.05) \[
M_1 = M_1^{(1)} M_2^{(1)} M_3^{(1)} = \begin{bmatrix}
\mathbf{I} & \mathbf{F}_{m+1}^{T} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\mathbf{I} & \mathbf{\theta} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\mathbf{I} & \mathbf{\theta} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\mathbf{I} & \mathbf{\theta} & \mathbf{0}
\end{bmatrix}
\]

If \( \mathbf{B}^{-1} = [\mathbf{P}_1 \ldots \mathbf{P}_m]^{-1} \) is retained in product form, then the required composite product form transformation is

(3.06) \[
\mathbf{T}_1 = M_1 \begin{bmatrix}
\mathbf{B}^{-1} & \mathbf{\theta}
\end{bmatrix}
\begin{bmatrix}
\mathbf{0}^T & 1
\end{bmatrix}
\]

It is easily verified that

(3.07) \[
\begin{bmatrix}
\mathbf{P}_j \\
-\alpha_{j,m+1}
\end{bmatrix} = \mathbf{u}(j) \quad (j = 1, \ldots, m)
\]

(3.08) \[
\begin{bmatrix}
\mathbf{P}_{m+1} \\
1
\end{bmatrix} = \mathbf{u}(m+1)
\]

(3.09) \[
\begin{bmatrix}
\mathbf{P}_j \\
0
\end{bmatrix} = \mathbf{P}_j \quad (j = m+2, \ldots, n).
\]
(Note that \( u(j), u^{(m+1)} \) and \( T_j \) are vectors with \( m+1 \) elements).

To compute \( M_1 y \) for some vector \( y \), one need have stored only the \( m+1 \) values \( e_{1,m+1} \) and \( e_{m+1} \).

When
\[
y = \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix} \begin{bmatrix} P_j \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{P}_j \\ 0 \end{bmatrix} \quad (j = m+2, \ldots, n)
\]
then
\[
\bar{T}_j = M_1 y = \begin{bmatrix} \bar{P}_j \\ 0 \end{bmatrix} + \frac{\bar{P}_{m+1} + \bar{P}_j}{1 + (\bar{P}_{m+1} \bar{P}_{m+1})} \begin{bmatrix} -\bar{P}_{m+1} \\ 1 \end{bmatrix},
\]
which follows from the representation of \( M_1 \) given below:
\[
M_1 = I - \frac{1}{\bar{P}_{m+1}} \begin{bmatrix} \bar{P}_{m+1} \bar{P}_{m+1} & \bar{P}_{m+1} \\ \bar{P}_{m+1} & \bar{P}_{m+1} \bar{P}_{m+1} \end{bmatrix}.
\]

The continuation of the process described above, yielding the vectors \( \bar{T}_{m+3}, \ldots, \bar{T}_n \), is performed using the transformation
\[
T_2 = M_2 \begin{bmatrix} T_1 \\ 0 \end{bmatrix}
\]
where
\[
M_2 = N_1(2) N_2(2) N_3 = \begin{bmatrix} I & -\bar{T}_{m+2} \\ \theta^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ \theta^T & \frac{1}{\bar{P}_{m+2}} \end{bmatrix} \begin{bmatrix} I & \theta \\ \theta^T & 1 \end{bmatrix} \begin{bmatrix} I & \bar{T}_{m+2} \\ \bar{T}_{m+2} & 1 \end{bmatrix}
\]
(3.13) \[ \sigma_{m+2} = 1 + (V_{m+2}, W_{m+2}) \]. In general, then, the transformation which yields the successive canonical forms is

(3.14) \[ T_{k+1} = M_{k+1} \begin{bmatrix} T_k & 0 \\ \theta^T & 1 \end{bmatrix} \text{ for } (k = 0, 1, 2, \ldots, (n-m-1)) \]

where

(3.15) \[ M_{k+1} = \begin{bmatrix} N_1^{(k+1)} & 0 & N_2^{(k+1)} & N_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} I & \frac{k+1}{F_{m+k+1}} \\ \theta^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 & \frac{1}{\sigma_{m+k+1}} \\ \theta^T & \frac{1}{\sigma_{m+k+1}} & \frac{k+1}{F_{m+k+1}} \end{bmatrix} \]

\[ \sigma_{m+k+1} = 1 + \frac{k+1}{F_{m+k+1}}, \]

\( T_0 = B^{-1} \) and \( F_{m+k+1} \) is the \((m+k+1)\text{st}\) column of the \((k+1)\text{st}\) canonical form. Note that \( T_{k+1} \) is an \((m+k+1) \times (m+k+1)\) matrix. The relations analogous to (3.07) are as follows:

\[
T_{k+1} = \begin{bmatrix} P_j \\ -\sigma_{j,m+1} \\ -\sigma_{j,m+2} \\ \vdots \\ -\sigma_{j,m+k+1} \end{bmatrix} = u(j) \quad (j = 1, 2, \ldots, m)
\]

(3.16) \[
T_{k+1} = \begin{bmatrix} P_{m+1} \\ 1 \\ -\sigma_{m+1,m+2} \\ -\sigma_{m+1,m+3} \\ \vdots \\ -\sigma_{m+1,m+k+1} \end{bmatrix} = u(m+1)
\]
where the unit vectors \( u(j) \) are the columns of the 
\((m+k+1) \times (m+k+1)\) identity matrix, \( \theta_k \) is a vector 
whose \( k \) elements are zeros, and 
\[
\begin{bmatrix}
p_j \\
\theta_{k+1}
\end{bmatrix}
= \frac{k+2}{p_j}
\text{ for } (j = m+k+2, \ldots, n).
\]

Consider the nonlinear programming problem

\[
(3.17) \quad \text{Maximize } g(x) \text{ where } F(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_t(x) \end{bmatrix} \geq 0.
\]

If \( g(x) \) and the \( f_i(x) \) \((i = 1, \ldots, t)\) are differentiable 
and concave (see [6],[2]) then if \( F(\hat{x}) \geq 0, \hat{x} \) is a
(global) constrained maximum of $g(x)$ if and only if there exists an $n$ by $r$ matrix of rank $r \leq p$ whose $r$ columns provide a set of linearly independent gradients to the $p$ constraints active at $\hat{x}$, such that

\begin{align}
(3.18) \quad & (I - N_p^T N_p^+) v \hat{g}(x) = 0 \\
(3.19) \quad & -N_p^+ v \hat{g}(\hat{x}) \geq 0.
\end{align}

A constraint $f_i(x)$ is active at $\hat{x}$ provided $f_i(\hat{x}) = 0$; otherwise $f_i(\hat{x}) > 0$ and $f_i(x)$ is said to be inactive at $\hat{x}$.

Rosen's conditions (3.18) and (3.19), qualified as above, are equivalent to the Kuhn-Tucker conditions [2]. The proof given in [2] requires only minor modifications in order to support the following:

\begin{align}
(3.20) \quad & \text{Let } N_p = [v f_{j_1}(\hat{x}) \ldots v f_{j_p}(\hat{x})] \\
& \text{be that matrix whose columns consist of the gradients to all $p$ active constraints. Then in (3.18) and (3.19) $r$ may be taken to be equal to the rank of $N_p$. Thus the matrix $N_r$ is composed as follows:} \\
(3.21) \quad & N_r = [v f_{j_1}(\hat{x}) \ldots v f_{j_r}(\hat{x})] \\
& \text{where } f_{j_1}(\hat{x}), \ldots, f_{j_r}(\hat{x}) \text{ form a subset of the active }
constraints, active at $\hat{x}$, and $r = \text{rank } N_p$.

Now restrict consideration to the linearly constrained, nonlinear programming problem

(3.22) \nMaximize $h(x)$ where $Ax = b$, $x \geq \theta$ ,

and $h(x)$ is concave, differentiable. In order to simplify the expositions we will continue to assume that $A$ is $m$ by $n$ of rank $m$ and, further, that all points in the set $\mathcal{A} = \{x | Ax = b, x \geq \theta \}$ are nondegenerate in the sense defined in [5]. A second paper will take up techniques for handling deficient rank and degeneracy, together with a related topic: the use of an interior gradient projection method [5] when some of the constraints $f_i(x)$, other than the non-negativity constraints $x \geq \theta$, are nonlinear.

If problem (3.22) is replaced by the equivalent problem

Maximize $g(z, x_1, \ldots, x_m) = z$

(3.23)

where

$h(x) - z = 0$

$Ax = b$

$x \geq \theta$
and each equality constraint is represented by a pair of inequality constraints, then at a non-degenerate vertex $\bar{x}$ of $A$, which corresponds to a canonical form

$$
\begin{pmatrix}
-1 & g^T & \bar{\sigma}_{m+1}(\bar{x}) & \cdots & \bar{\sigma}_n(\bar{x}) & -\bar{z}_o \\
0 & I & \bar{p}_{m+1} & \cdots & \bar{p}_n & \bar{p}_{o}
\end{pmatrix}
$$

(3.24)

where

$$\bar{z}_o = h(\bar{x})$$

and the $\bar{\sigma}_j(\bar{x})$ are obtained by transforming the vector $vh(\bar{x})$ in the standard way [1], the matrix $N_p$ is composed as follows:

$$
\begin{pmatrix}
-1 & g^T \\
-\bar{\sigma}_{m+1} & \bar{p}_{m+1}^T \\
\vdots & \vdots \\
-\bar{\sigma}_n & \bar{p}_n^T 
\end{pmatrix}
$$

$$
\begin{pmatrix}
-1 & g^T \\
-\bar{\sigma}_{m+1} & \bar{p}_{m+1}^T \\
\vdots & \vdots \\
-\bar{\sigma}_n & \bar{p}_n^T 
\end{pmatrix}
$$

$$
\begin{pmatrix}
0 \\
\bar{p}_{m+1}^T \\
\vdots \\
\bar{p}_n^T 
\end{pmatrix}
$$

(3.25)

Since $\bar{x}$ non-degenerate implies rank $N_p = n+1$, it follows that the matrix $N_r$ of (3.18), (3.19) consists of $(n+1)$ columns; precisely $(m+1)$ of the columns of $N_r$ are taken from the first two submatrices of $N_p$, the remainder being taken from the third submatrix. If column $j$ appears in $N_r$, column $(j+m+1)$ does not, and conversely, for
\( j = 1, 2, \ldots, m + 1 \). In any event, \( N_{\tau} \) is nonsingular
and thus
\[
I - N_{\tau}N_{\tau}^+ = I - N_{\tau}N_{\tau}^{-1} = I - I = 0
\]

and (3.18) is trivially satisfied. Put somewhat
differently, the vertex \( \bar{x} \) of \( A \) lies in the 0-dimensional
intersection of the active constraints.

Condition (3.19) is realized by solving the system

\[
(3.26) \quad N_{\tau}w = u(1) = v_g
\]

(\( u(1) \) here has \( (n+1) \) elements)

for its unique solution \( w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \), as follows:

Case A: If column 1 of \( N_p \) appears in \( N_{\tau} \),
\[
w_0 = -1, \quad w_1 = \ldots = w_m = 0, \quad w_{m+1} = \bar{\tau}_{m+1}, \ldots, w_n = \bar{\tau}_n.
\]

Case B: If column \( m+2 \) of \( N_p \) appears in \( N_{\tau} \),
\[
w_0 = 1, \quad w_1 = \ldots = w_m = 0, \quad w_{m+1} = \bar{\tau}_{m+1}, \ldots, w_n = \bar{\tau}_n.
\]

Then we have

Theorem 3.1: The non-degenerate \( \bar{x} \) is a maximizing
solution for (3.23) if and only if
\[(3.27) \quad \overline{r}_j(x) \leq 0 \quad (j = m+1, \ldots, n), \text{ where the } \overline{r}_j(x)\]

are defined in (3.24).

**Proof:** If \( x \) is a maximizing solution for (3.24), then (3.18) and (3.19) imply that the solution \( w \) of (3.26) is such that \( w \leq 0 \). This means that Case B above is impossible, thus Case A must hold, in which case \( \overline{r}_j(x) \leq 0 \) for \( j = m+1, \ldots, n \).

If \( \overline{r}_j(x) \leq 0 \) for \( j = m+1, \ldots, n \), then with \( w \) determined as in Case A, conditions (3.18) and (3.19) hold with any \( N_p \) (of the type specified following relation (3.25)) having as its first column the first column of \( N_p \) in (3.25). Thus \( x \) is a maximizing solution for (3.23).

At a vertex of \( A \) the condition

\[ w = N^*_r v \leq 0 \]

is thus the well-known condition involving the "relative cost factors" \( \overline{r}_j \) [1].

If instead of (3.22) the problem is formulated as

\[(3.28) \quad \text{Minimize } f(x) \text{ where } Ax = b, x \geq 0, \text{ where } f(x) \text{ is convex, differentiable, then the above conditions become those for maximizing } -f(x) \text{ and we take } h(x) = -f(x), \text{ in which case, assuming the } \overline{r}_j(x)\]

values are obtained at a vertex \( x \) by suitably transforming the vector \( v(x) \), condition (3.19) becomes
\[-\bar{\sigma}_j(\bar{x}) \leq 0; \text{ that is,}\]

\[(3.29) \quad \bar{\sigma}_j(\bar{x}) \geq 0 \quad \text{for } (j = m+1, \ldots, n).\]

Since the developments in sections §1 and §2 follow Dantzig's practice of formulating problems in terms of minimization, whereas Rosen formulates problems in terms of maximization, it is impossible to continue notational agreement. In the remainder of the paper we will follow Dantzig's practice. Thus suppose Rosen's gradient projection method [6] is initiated at a vertex \(\bar{x}\) of \(\Lambda\). Such a vertex could be obtained, for example, by executing "Phase I" of the simplex method [1]. Then, unless \(\bar{x}\) is an optimal solution for the problem:

\[(3.30) \quad \text{Minimize } f(x) \text{ where } Ax = b, x \geq 0, f(x) \text{ convex, differentiable;} \quad \text{it must be that } \bar{\sigma}_j(\bar{x}) < 0 \text{ for at least one value of } j. \text{ Defining } s, \text{ which for convenience we label m+1, by letting}\]

\[\bar{\sigma}_s(\bar{x}) = \min \{ \bar{\sigma}_j(\bar{x}), \quad \bar{\sigma}_j(\bar{x}) < 0 \}\]

compute the direction of a vector \(\eta^{(m+1)}\) along the associated edge of \(\Lambda\), determining a path in that direction. Let \(\tilde{x}\) be that feasible point located at the maximal distance from \(\bar{x}\) along this (linear) path. Then determine \(\hat{x}\) such that \(f(\hat{x}) \leq f(x)\) for
all $x = \eta \bar{x} + (1-\eta) \tilde{x}$ where $0 \leq \eta < 1$. If $\hat{x} = \bar{x}$, the process then employed is a simplex iteration. If $\hat{x} \neq \bar{x}$, then at $\hat{x}$

$$
\eta^{(m+1)} \eta^{(m+1)^T} \eta f(\hat{x}) = 0,
$$

since otherwise $x$ may be determined on the path with $f(\hat{x}) < f(\bar{x})$. Thus $\eta f(\hat{x})$ is orthogonal to the vector $\bar{x} - \tilde{x}$. Note that, as usual, a 1-dimensional minimization problem must be solved; we assume that this presents no difficulties. The point $\hat{x}$ lies in the 1-dimensional intersection of the active constraints, $A x = b$, $x_m+2 = \ldots = x_n = 0$, precisely at the point where it intersects the hyperplane $(\eta^{(m+1)}, x) = (\eta^{(m+1)}, \hat{x})$.

Let

$$
\eta = \begin{bmatrix}
-1 & \eta^{T} \\
-\bar{x}^{T} & \eta^{T} \\
\bar{x}^{T} & \eta^{T} \\
\eta^{T} & \eta^{T} \\
\eta^{T} & \eta^{T} \\
\eta^{T} & \eta^{T} \\
\eta^{T} & \eta^{T}
\end{bmatrix}
\begin{bmatrix}
0 \\
I
\end{bmatrix}
$$

where the values $\bar{c}_j(\hat{x})$ for $(j = m+2, \ldots, n)$ are obtained by transforming the vector $\eta f(\hat{x})$. (A method for determining the successive vectors of $\bar{c}_j$ values, using a variation of the simplex algorithm with multipliers, is given at the end of this section.) Then we have
Theorem 3.2: If \( \eta^{(m+1)}\eta^{(m+1)} \varphi(x) = 0 \) then \( \hat{x} \) is a minimizing solution of (3.28) if and only if \( \overline{c}_j(\hat{x}) \geq 0 \) for \( (j = m+2, \ldots, n) \).

Proof: Suppose \( \overline{c}_j(\hat{x}) < 0 \) for some value \( j_0 \) in the index set \( (j = m+2, \ldots, n) \). Then \( \hat{x} \) is not a minimizing solution of (3.28) since \( f(x) \) can be decreased by treating \( \hat{x} \) as a vertex of the convex set

\[
A^{(1)} = \{x | Ax = b; (x, \eta^{(m+1)}) = (\hat{x}, \eta^{(m+1)}) \geq \overline{c}_j \}
\]

by proceeding along an edge of \( A^{(1)} \) determined by \( j_0 \). Thus \( \hat{x} \) a minimizing solution of (3.28) implies \( \overline{c}_j(\hat{x}) \geq 0 \) for \( (j = m+2, \ldots, n) \).

Now suppose that \( \overline{c}_j(\hat{x}) \geq 0 \) for \( (j = m+2, \ldots, n) \), and that \( \hat{x} \) is not a minimizing solution for (3.28).

Let

\[
M_r = \begin{bmatrix}
1 & \theta^T & \cdots & \theta^T \\
\varphi(\hat{x}) & A^T & u^{(m+2)} & \ldots & u^{(n)}
\end{bmatrix}
\]

and \( A = \begin{bmatrix}
a^{(1)} \\
\vdots \\
a^{(m)}
\end{bmatrix} \) where \( a^{(1)} \) is row 1 of \( A \).

Since \( \eta^{(m+1)}\eta^{(m+1)}T \varphi(\hat{x}) = 0 \), \( \varphi(\hat{x}) \in M(\eta^{(m+1)}) \).
But \( \mathfrak{m}(n(m+1)) = (\cap_{i=1}^{n} \mathfrak{m}(u(1))) \cap \mathfrak{m}(a(1)) \cap (\cap_{i=m+2}^{n} \mathfrak{m}(u(1))) \cap \mathfrak{m}(a(1)) \cap \cdots \cap \mathfrak{m}(a(m)) \cap \cdots \cap \mathfrak{m}(u(n)), \)

thus there exists a (unique) solution \( v \) to the system

\[
[\mathbf{A}^T \ u^{(m+2)} \ \cdots \ u^{(n)}] \ v = \mathbf{v}_f(\mathbf{x})
\]

where \( v = \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} \), and we have \( \mathbf{N}_r \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{v}_g = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \)

where \( \mathbf{N}_r \) is defined in (3.33). Sure \( \mathbf{N}_r \) has full column rank,

\[
\mathbf{N}_r^T = (\mathbf{N}_r^T \mathbf{N}_r)^{-1} \mathbf{N}_r^T
\]

hence

\[
\mathbf{N}_r^T \mathbf{v}_g = \begin{bmatrix} 1 \\ -v \end{bmatrix}
\]

thus

\[
\mathbf{N}_r^T \mathbf{N}_r \mathbf{v}_g = \mathbf{v}_g,
\]
that is,

\[(I - \mathbf{N}_r \mathbf{N}_r^T) \mathbf{v}_g = 0.\]

Therefore, if \( \hat{x} \) is not a minimizing solution for (3.28), \( v \) must have at least one element \( v_{k_0} \), \( k_0 \) in the index set \( (k = 1, \ldots, n-1) \), such that \(-v_{k_0} < 0\), i.e. such that \( v_{k_0} > 0 \).

Now let

\[\mathbf{N}_p = \begin{bmatrix}
-1 & \mathbf{g}_T \\
-\mathbf{v}_T(\hat{x}) & \mathbf{A}_T^T
\end{bmatrix}, \quad
\begin{bmatrix}
-1 & \mathbf{g}_T \\
-\mathbf{v}_T(\hat{x}) & \mathbf{A}_T^T
\end{bmatrix}^T, \quad
\begin{bmatrix}
-\mathbf{g}_T \\
\mathbf{u}_{(m+2)}^{(m+2)}, \ldots, \mathbf{u}(n)
\end{bmatrix}.
\]

It follows that the conclusion of the preceding paragraph may be drawn for any \( \mathbf{N}_r \) composed of \( n \) linearly independent columns taken from \( \mathbf{N}_p \). As in the development following relation (3.25), precisely \( (m+1) \) of the columns of such an \( \mathbf{N}_r \) are taken from the first two submatrices of \( \mathbf{N}_p \), the remaining columns being taken from the third. If column \( j \) appears in \( \mathbf{N}_r \), column \((j+m+1)\) does not, and conversely, for \((j = 1, 2, \ldots, m+1)\), \( \mathbf{N}_r \) here has rank \((n-1)\).

By previous developments, \( \mathbf{a}_j(\hat{x}) \geq 0 \) for \((j = m+2, \ldots, n)\) implies that \( \hat{x} \) is a minimizing solution for the problem

\[\text{Minimize } f(x) \text{ where } \begin{align*}
\mathbf{A}x &= \mathbf{b} \\
\mathbf{\eta}(m+1) &\mathbf{x} = (\eta_{(m+1)}, \hat{x}) \\
\text{and} & \quad x \geq 0.
\end{align*}\]
Thus, defining

\[
\tilde{N}_p = \begin{bmatrix}
-1 & \Theta^T & 0 \\
-vf(\tilde{x}) A^T \eta^{(m+1)} & \vdots & -1 & \Theta^T & 0 \\
& -vf(\tilde{x}) A^T \eta^{(m+1)} & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\]

Then (3.18) and (3.19) imply that there exists a matrix \( \tilde{N}_r \) of rank \((n+1)\) with columns chosen as in the development following relation (3.25) (except that \((m+2)\) columns are taken from the first two submatrices; as before, \( \tilde{N}_r \) is nonsingular, thus (3.19) is trivially satisfied), and such that the unique solution \( z \) of

\[
\tilde{N}_r z = v_g,
\]

satisfies

\[
z \geq 0, \quad \text{where } z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_n \end{bmatrix}.
\]

Since \( z_0 = 1 \), it follows that \( z_0 = 1 \), thus that column \((m+3)\), \([vf(\tilde{x})]\), of \( \tilde{N}_p \) appears in \( \tilde{N}_r \). Designating \( \tilde{N}_r \) as follows:

\[
\tilde{N}_r = \begin{bmatrix}
1 & \Theta^T & \cdots & \cdots & \cdots \\
-vf(\tilde{x}) D & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
& \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\]

where \( D = [d^{(1)} \ldots d^{(m+1)}] \) is composed of \((m+1)\) linearly independent columns taken from \([A^T \eta^{(m+1)}] \) and \([-A^T \eta^{(m+1)}] \), thus

\[
z_0 vf(\tilde{x}) + [D, u^{(m+2)} \ldots u^{(n)}] \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = 0,
\]
where \( z_i \geq 0 \) for \( i = 1, 2, \ldots, n \) and \( z_0 = l \), or

\[
[D, \ u^{(m+2)} \ldots \ u^{(n)}] \begin{bmatrix}
-z_1 \\
\vdots \\
-l \end{bmatrix} = v_f(\hat{x})
\]

where \( -z_i \leq 0 \). Now in reference to (3.33), take

\[
N' = \begin{bmatrix}
1 & \ldots & \ldots & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \ldots & \ldots & 1 & 0 \\
v_f(\hat{x}) & D' & u^{(m+2)} & \ldots & u^{(n)}
\end{bmatrix}
\]

where \( D' \) consists of the first \( (m+1) \) columns of \( D \). As pointed out following relation (3.37), \( (I-(N'_r)(N'_r)^+)^v_g = 0 \)
and if \( \hat{x} \) is not a minimizing solution for (3.26), the associated \( v' \), which solves the system analogous to

(3.34), must have at least one element \( v'_{k_o} > 0 \). The

uniqueness of the representation of \( v_f(\hat{x}) \) as a linear combination of the columns of \([D, u^{(m+2)} \ldots u^{(n)}]\) then

implies \(-z_1 = v'_1, -z_2 = v'_2, \ldots , -z_m = v'_m, -z_{m+1} = 0, -z_{m+2} = v'_{m+1}, \ldots , -z_n = v'_{n-1} \) and \( v'_{k_o} > 0, -z_1 \leq 0 \)

provides a contradiction, thus \( \hat{x} \) is a minimizing

solution of (3.28).

Let us review:

Starting at a vertex \( \hat{x} \) of \( \Lambda = \{x | Ax = b, x \geq 0\} \), a path

along an edge of \( \Lambda \) is determined. This path is followed

to the point \( \hat{x} \), possibly an adjacent vertex of \( \Lambda \). If

that be the case, the process employed is the simplex

algorithm. If \( \hat{x} \) is not a vertex of \( \Lambda \), then the values
\( \bar{\sigma}_j(\hat{x}) \) are determined and if \( \bar{\sigma}_j(\hat{x}) \geq 0 \) for \( j = m+2, \ldots, n \), 
\( \hat{x} \) is optimal for (3.28). Otherwise \( \hat{x} \), a vertex of 

\[
\Lambda^{(1)} = \{ x | Ax = b, (\eta^{(m+1)}, x) = (\eta^{(m+1)}, \hat{x}), x \geq 0 \}
\]

is treated in the same fashion as was \( \hat{x} \), with the 
following exception: In general, a path along an edge 
of \( \Lambda^{(1)} \) (in the feasible direction of the vector \( \eta^{(m+2)} \)) 
developed in §2) lies on a 2-dimensional face of \( \Lambda \). 
If a point \( \hat{x} \), not a vertex of \( \Lambda^{(1)} \), is determined on 
such a path, then \( \eta^{(m+2)}(\eta^{(m+2)})^T \bar{v}(\hat{x}) = 0 \) as before, 
however this need not imply that \( \bar{v}(\hat{x}) = 0 \), where 
\( P = \eta^{(m+1)}(\eta^{(m+1)})^T + \eta^{(m+2)}(\eta^{(m+2)})^T \). At such a point 
one first determines whether or not \( \bar{v}(\hat{x}) = 0 \). 
If \( \bar{v}(\hat{x}) \neq 0 \), a new 1-dimensional path on the related 
2-dimensional face of \( \Lambda \) is followed. (Note that if 
\( \hat{x} \) is a vertex of \( \Lambda^{(1)} \), the process employed is the 
simplex algorithm.) The procedure outlined above is 
continued until one of two situations occurs:

\[
(3.42)
\]

The terminal point, \( \bar{x} \), lies on the related 2-dimensional 
face of \( \Lambda \), no additional non-negativity 
constraints having become active, and is such that 

\[
\bar{v}(\bar{x}) = 0 \text{ where } P = \eta^{(m+1)}(\eta^{(m+1)})^T + \eta^{(m+2)}(\eta^{(m+2)})^T
\]
or
In a manner completely analogous to that already discussed, it is apparent that the proof of Theorem 3.2 need be modified only slightly to support the indicated generalization; both its formulation and the modified proof are left to the reader.

The generalization of this approach to higher dimensional faces of A is quite apparently analogous to our discussion of 0, 1, and 2 dimensional faces. Since it seems likely that the interesting hyperplanes at such a point will normally include many of those which intersect the last vertex of A used in
the process. Thus if the $B^{-1}$ associated with this 
last vertex has been retained, as would usually be
the case, an appropriate vertex, "adjacent" to the
current point may be obtained by application of a few
"Phase I" simplex steps.

To obtain the $\bar{c}_j(\hat{x})$ values using the method of
simplex multipliers [1], one solves, for example,

\begin{equation}
\pi T_1^{-1} = \gamma = [\bar{c}_1(\hat{x}) \ldots \bar{c}_{m+1}(\hat{x})]
\end{equation}

for the row vector

\begin{equation}
\pi(\hat{x}) = [\pi_1(\hat{x}) \ldots \pi_{m+1}(\hat{x})]
\end{equation}

where $\bar{c}_j(x) = \frac{\partial r(\hat{x})}{\partial x_j}$ for $(j = 1, \ldots, m+1)$. Then the
$\bar{c}_j(\hat{x})$ values for $(j = m+2, \ldots, n)$ are obtained from the
relation

\begin{equation}
\bar{c}_j(\hat{x}) = c_j(\hat{x}) - \pi(\hat{x}) \left[ \begin{array}{c} p_j \\ 0 \end{array} \right].
\end{equation}

From (3.44) (see (3.12)) we have

\begin{equation}
\left[e_1 a_2 \ldots a_{m+1}\right] N_1^{(1)} N_2^{(1)} N_3^{(1)}
= \left[c_1 + a_{1,m+1} a_{m+1} + \ldots + a_{m,m+1} a_{m+1}^m\right]
\end{equation}

where

\begin{equation}
c_{m+1} = \frac{\sum_{i=1}^{m} a_i c_i}{c_{m+1}}.
\end{equation}
Thus

\[(3.48) \quad \begin{bmatrix} a_1 & a_2 & \cdots & a_{m+1} \end{bmatrix} T_i = \begin{bmatrix} e_1 & e_2 & \cdots & e_m \end{bmatrix} B^{-1} a_{m+1} B^{-1} c_{m+1} = [\pi_1 \pi_2 \cdots \pi_{m+1}] \]

A similar relation may be developed for subsequent transformations \( T_{k+1} \).

**Concluding remark:** In the approach given here the simplex algorithm is used in generating the projection matrix \( I - N^T N_q q \), whereas Rosen gives a method for obtaining \( N^T N_q q \) [6] based upon an algorithm involving \( (N^T N_q q)\)^{-1}. Since it appears that \( \text{dim } \mathcal{R}(I - N^T N_q q) \) will usually be much smaller than \( \text{dim } \mathcal{R}(N^T N_q q) \), one might expect significant computational improvements. This expectation is further enhanced by the potential use of a variation of the product form of the inverse in computing the vectors which constitute the required projection and by the use of "simplex multipliers" and "relative cost factors" [1] in the standard fashion of linear programming algorithm technology.
REFERENCES


