

SOLVING LINEARLY CONSTRAINED
NONLINEAR PROGRAMMING PROBLEMS
BY NEWTON'S METHOD

by

Fatemeh Ghotb

Serial No. 30, 1986

Faculty of Business Staff Papers



SWINBURNE INSTITUTE OF TECHNOLOGY

A division of Swinburne Ltd

SOLVING LINEARLY CONSTRAINED
NONLINEAR PROGRAMMING PROBLEMS
BY NEWTON'S METHOD

by

Fatemeh Ghotb

Serial No. 30, 1986

ISBN 0 85590 586 5

This paper should not be quoted or reproduced in whole or in part without the consent of the author, to whom all comments and enquiries should be directed.



Ghotb, F., 1986

ABSTRACT

In this paper matrix techniques are developed to implement a modified Newton method to a linearly constrained nonlinear optimisation problem with a twice differentiable and factorable objective function.

The problem first is reduced to an unconstrained problem, then the generalised inverse of the positive part of the Hessian matrix is used to generate the direction of search.

1. Introduction

This paper is concerned with solving the following linearly constrained nonlinear programming problem:

$$\text{minimize } f(x) \quad (1.1)$$

$$x \in \Omega$$

$$\text{subject to } Nx = e \quad (1.2)$$

$$Ax \geq b \quad (1.3)$$

where Ω is an open set, N is an $\ell \times n$ matrix of rank ℓ , A is an $m \times n$ matrix, $b \in E^m$, $e \in E^\ell$ and $f(x)$ is twice continuously differentiable (i.e. $f \in C^2$).

The point of view taken here is that at each iteration, the above **problem** can be converted to an unconstrained minimization of $f(x)$ in a **subspace** of reduced dimension which can be solved by several methods. The unconstrained minimization technique used here is the following modified Newton method:

$$x^{k+1} = x^k - \alpha^k [\nabla_x^2 f(x^k)]^{-1} \nabla_x f(x^k)$$

The **parameter** α^k is the step size and is chosen from optimal step size procedure discussed in section 8 to enforce $f(x^{k+1}) < f(x^k)$.

In this paper formulas are developed for the matrix methods required to implement the modified Newton method in factorable form. These are two types. First, are the formulas required for the computation of a nullspace matrix, a matrix whose columns span the space of vectors orthogonal to the derivatives of active constraints. Two approaches, elimination of variables and the projection method are examined. The second set of matrix computations are those required to estimate the generalized inverse of the "positive part" of the Hessian matrix in the equality constrained subspace [8]. $f(x)$ also is assumed to be a factorable function [11], for which a dyadic form Hessian matrix can be formed and its generalized inverse can be updated by rank one matrices at each iteration.

2. Equivalence of Linearly constrained and unconstrained problems.

Consider the problem in 1.1 - 1.3. At any feasible point $\mathbf{x}_S$ let $\hat{R}$ be the set of indices i for which $\mathbf{a}_i \mathbf{x}_S = \mathbf{b}_i$ (from $i=1, \dots, m$), and assume that the vectors in the set

$$S = \{N^T, \{a^T\}, i \in \hat{R}\} \quad (2.1)$$

are linearly independent. Since at any stage of the algorithm only the equality and active constraints take part, we can define the following problem

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}) \\ \text{Subject to} & C \mathbf{x} = \mathbf{d} \end{array} \quad (2.2)$$

where C is a $r \times n$ matrix, $r \leq n$, it has full rank, and $C_i \in S$, $i=1, \dots, r$ (C_i is the i th row of the matrix C). If x_s is a point satisfying the constraints in (2.2) then a necessary and sufficient condition that some point x also satisfies $Cx = d$ is that

$$x = x_s + y$$

where y is in the nullspace of C .

Let B be an $n \times (n-q)$ matrix such that a necessary and sufficient condition for a vector y to be in the null space of C is,

$$y = Bz \tag{2.3}$$

for some $z \in E^{n-q}$. Then it follows that the problem (2.2) can be converted into the following unconstrained problem:

$$\text{Minimize } F(z) = f(x_s + Bz) \tag{2.4}$$

$$z \in E^{n-q}$$

A sequence x^k tending to solve (2.2) can be regarded as being generated by a sequence z^k which attempts to solve (2.4). It is shown in [12] that a necessary **and** sufficient condition for a point x^k satisfying the set of constraints $Cx^k = d$, to be a **Lagrangian** stationary point is that

$$\nabla_z F(z) = 0$$

It is also shown that if

$$B^T \nabla f(x^k) B$$

is a positive semi-definite matrix, i.e.

$$z^T B^T \nabla^2_x f(x^k) B z \geq 0 \text{ for all } z \in E^{n-q},$$

then x^k satisfies the second order necessary conditions as well.

3. Null Space Computation

To solve the unconstrained problem (2.4), the following iterative formula is used.

$$z^{k+1} = z^k - \alpha^k [\nabla_{zz}^2 F(z^k)]^+ \cdot \nabla_z F(z^k) \quad (3.1)$$

premultiplying by B yields

$$x^{k+1} = x^k - \alpha^k B [\nabla_{xx}^2 f(x^k) B]^+ [B^T \nabla f(x^k)] \quad (3.2)$$

where (+) stands for the generalized inverse.

In order to be able to use (3.2), we need to calculate the nullspace matrix of the matrix C, the matrix of the gradient of all binding constraints. Two methods, Elimination of variables and projection method are considered. The following theorem is the basis for working with different methods of nullspace matrix computation.

Theorem

Let B_1 and B_2 be two matrices generating the null space of C (defined as in (2.2)), with $n \times q_1$ and $n \times q_2$ dimensions respectively and $q_1, q_2 \geq (n-r)$. Furthermore, suppose $z^T \nabla^2 f(x) z > 0$ for all $z \neq 0$ and $Cz=0$. Then it is proved that [12]:

$$B_1 (B_1^T \nabla^2 f(x) B_1)^+ B_1^T = B_2 (B_2^T \nabla^2 f(x) B_2)^+ B_2^T \quad (3.3)$$

Now let's reconsider (2.2) where the matrix C is partitioned as (C_D, C_I) and C_D is an $r \times r$ matrix of full rank and C_I an $r \times (n-r)$ matrix. The vector x is also partitioned as $\begin{pmatrix} x_D \\ x_I \end{pmatrix}$, where x_D

and x_I have r and $(n-r)$ components respectively. Then the constraints in (2.2) can be written as:

$$(C_D, C_I) \begin{pmatrix} x_D \\ x_I \end{pmatrix} = d$$

$$x_D = (C_D)^{-1}d - (C_D)^{-1}C_I \cdot x_I, \quad (3.4)$$

and the linearly constrained problem (2.2) can be converted to the following unconstrained minimization.

$$\begin{aligned} \text{Minimize} \quad & f(x_D(x_I), x_I) \\ & x_I \in E^{n-r} \end{aligned} \quad (3.5)$$

Comparing (3.5) with (2.4). x_I plays the role of z and therefore the null space matrix is

$$B = \begin{pmatrix} -(C_D)^{-1} C_I \\ I \end{pmatrix} \quad (3.6)$$

where I is an $(n-r)$ identity matrix. The estimates of the **Lagrange** multipliers of the binding constraints at k th iteration would be as follows:

$$w_k = (C_D^{-T}) \nabla_D f(x^k) \quad (3.7)$$

Similarly, applying the projection method the null space matrix will be equal to the projection matrix P given as:

$$P = B = I - C^T(CC^T)^{-1}C \quad (3.8)$$

With the estimates of **Lagrange** multipliers at k th iteration given by:

$$w^k = (CC^T)^{-1} C \nabla f(x^k) \quad (3.9)$$

4. Null space matrix Updating

At the beginning of the first step of the k^{th} iteration and also the beginning of each step, it might happen that one of the inequality constraints becomes active or an active constraint leaves the boundary and becomes inactive. In either of these two cases the null space matrix must be updated.

When the projection method is used for null space matrix calculation it is inefficient to calculate the projection matrix explicitly. Thus, the only thing we keep is $(CC^T)^{-1}$ and we have to update it whenever any of the above mentioned cases occurs. The general formula for the inverse of a symmetric bordered matrix is

$$\begin{pmatrix} B & b^T \\ b & g \end{pmatrix}^{-1} = \begin{pmatrix} B^{-1} + B^{-1}b^T\alpha bB^{-1} & -B^{-1}b^T\alpha \\ -\alpha bB^{-1} & \alpha \end{pmatrix} \quad (4.1)$$

where $\alpha = (-bB^{-1}b^T + g)^{-1}$, and B is a symmetric matrix.

As a new constraint's boundary is encountered, this amounts to inverting

$$\begin{pmatrix} C \\ a \end{pmatrix} (C^T, a^T) = \begin{pmatrix} CC^T & Ca^T \\ aC^T & aa^T \end{pmatrix} \quad (4.2)$$

Knowing $(CC^T)^{-1}$ we can make the identification of (CC^T) with B , aC^T with b , aa^T with g and use the general formula (4.1) above.

If a constraint's boundary is relinquished, a reverse operation must be performed, thus having the inverse of (4.2) it is required to find $(cc^T)^{-1}$, suppose it is given that,

$$\begin{pmatrix} B & b^T \\ b & t \end{pmatrix}^{-1} = \begin{pmatrix} D & d^T \\ d & g \end{pmatrix} \quad (4.3)$$

Then

$$B^{-1} = D - d^T g^{-1} d$$

The general formula for the inverse of a bordered non-symmetric matrix which is used to update the null space matrix in the case of elimination of variable approach is as follows:

$$\begin{pmatrix} B & e^T \\ b & t \end{pmatrix}^{-1} = \begin{pmatrix} B^{-1} + B^{-1} e^T \alpha b B^{-1}, & -B^{-1} e^T \alpha \\ -\alpha b B^{-1} & , & \alpha \end{pmatrix}$$

$$\text{where } a = (-b B^{-1} e^T + t)^{-1} \quad (4.4)$$

Equation (4.4) is used when a constraint's boundary is encountered.

5. Computing the positive part of the Hessian

Recall the problem **(2.4)**. If $\mathbf{f}$ is a twice continuously differentiable and factorable function then the natural way of writing the Hessian in factorable form $[g]$ is:

$$\nabla^2 \mathbf{f}(\mathbf{x}) = \sum_{j=1}^L \mathbf{a}_j^k \mathbf{c}_j^k (\mathbf{a}_j^k)^T \quad (5.1)$$

If we divide the set of indices $j, j = 1 \dots, L$ such that

$$R = \{j | \mathbf{c}_j^k > 0\} , \quad (5.2)$$

$$S = \{j | \mathbf{c}_j^k < 0\}$$

then

$$\nabla^2 \mathbf{f}(\mathbf{x}) = \sum_{j \in R} \mathbf{a}_j^k \mathbf{c}_j^k (\mathbf{a}_j^k)^T + \sum_{j \in S} \mathbf{a}_j^k \mathbf{c}_j^k (\mathbf{a}_j^k)^T \quad (5.3)$$

resulting the following dyadic form for $\nabla^2 \mathbf{F}(\mathbf{z})$

$$\nabla^2 \mathbf{F}(\mathbf{z}) = \sum_{j \in R} \mathbf{B}^T \mathbf{a}_j^k \mathbf{c}_j^k (\mathbf{B}^T \mathbf{a}_j^k)^T + \sum_{j \in S} \mathbf{B}^T \mathbf{a}_j^k \mathbf{c}_j^k (\mathbf{B}^T \mathbf{a}_j^k)^T$$

Now let $(\mathbf{A}^k)$ be the estimate of the positive part of $\nabla_{\mathbf{z}\mathbf{z}}^2 \mathbf{F}(\mathbf{z})$. If $\nabla_{\mathbf{z}\mathbf{z}}^2 \mathbf{F}(\mathbf{z})$ is positive semi-definite, $\mathbf{A}$ will be equal to it. Otherwise, $\mathbf{A}$ is an overestimate in that the positive eigenvalues of the positive part are **pairwise** smaller than the corresponding positive eigenvalues of $\mathbf{A}$.

Also let $(A^k)^+$ be the generalized inverse of A^k . The algorithm starts at $j = 0$ with $(A^k)^+ = 0$ and perturbs A_j^k with a new dyad $B_{j+1}^{T a_j^k} c_{j+1} (B_{j+1}^{T a_j^k})^T$. At first all $j \in R$ are absorbed one by one, then the algorithm absorbs as many dyads $j \in S$ as possible, still maintaining the positive semi-definiteness of A . At this point $(A^k)^+$ will be used to generate the Newton search vector;

$$S = -BA^+\nabla_z F(z) = -BA^+B^T\nabla f(x). \quad (5.5)$$

6. Updating the projected Hessian

The generalized (Penrose-Moore) inverse of a matrix A is the unique matrix satisfying the following four properties (see [5], [28] for proofs and uniqueness and existence of the generalized inverse):

$$AA^+A = A \quad (6.1)$$

$$A^+AA^+ = A^+ \quad (6.2)$$

$$(AA^+)^T = (AA^+) \quad (6.3)$$

$$(A^+A)^T = A^+A$$

when A has an inverse $A^+ = A^{-1}$.

Let a be a one by n derivative vector of the constraint whose boundary was just encountered. Now $P = I - c^T(cc^T)^{-1}c$ is the projection matrix used to generate the null space of the matrix of binding constraints gradients.

And let

$$A = P \nabla^2 f(x) P$$

denote the projected Hessian, and $\hat{P}$ be the new projection matrix which is

$$\hat{P} = I - (C^T, a^T) \left\{ \begin{pmatrix} C \\ a \end{pmatrix}^T (C^T, a^T) \right\}^{-1} \begin{pmatrix} C \\ a \end{pmatrix}.$$

What is wanted now is an efficient iterative procedure to obtain

$$(\hat{A})^+ = (\hat{P} \nabla^2 f(x) \hat{P})^+ \quad (6.5)$$

and by efficient we mean using previous information as much as possible.

Note that $\hat{P}$ can also be written as:

$$\hat{P} = P - P a^T (a P a^T)^{-1} a P. \quad (6.6)$$

There are two cases to be considered:

Case I:

If $[P - A^+ A] a^T = [P - (P \nabla^2 f(x) P)^+ (P \nabla^2 f(x) P)] a^T = 0$,

then it is proved in [24] that,

$$(\hat{A})^+ = A^+ - A^+ a^T (a A^+ a^T)^{-1} a A^+ \quad (6.7)$$

Case II

If, $[P-AA^+]\mathbf{a}^T = 0$
 then (6.5) can be written

$$\begin{aligned}\hat{A} = A + [P\mathbf{a}^T - A\mathbf{a}(\alpha\beta)^{-1}](\alpha^2\beta)[\mathbf{a}P - (\alpha\beta)^{-1}\mathbf{a}A] \\ + A\mathbf{a}^T(-1/\beta)\mathbf{a}A\end{aligned}\quad (6.8)$$

where for the sake of simplicity $\mathbf{a} = (\mathbf{a}P\mathbf{a}^T)^{-1}$, and
 $\beta = (\mathbf{a}A^+\mathbf{a}^T)$

The closed form solution for this case is very complicated [24].

It should be noted that the projected gradient of the objective function should also be updated;

$$\hat{P}\nabla f(\mathbf{x}) = (P\nabla f) - P\mathbf{a}^T(\mathbf{a}P\mathbf{a}^T)^{-1}\mathbf{a}(P\nabla f) \quad (6.9)$$

7. Updating The Reduced Hessian

Suppose the matrix of the gradients of the binding constraints takes the form

$$C = (C_D, h^T, E)$$

where C_D is an r by r , h is 1 by r and E is an r by $(n-r-1)$ matrix.

The matrix generating the null space for the binding constraints gradients, takes the form

$$\begin{pmatrix} -C_D^{-1}h^T & -C_D^{-1}E \\ & I \end{pmatrix} \quad (7.1)$$

where I has rank $(n-r)$.

Let (c, g, e) be the gradient of a constraint just encountered where c is 1 by r , g is scalar, e is 1 by $(n-r-1)$.

The new null space matrix will take the form,

$$\begin{pmatrix} - \begin{pmatrix} C_D^{-1} h^T \\ c \end{pmatrix} \\ \tilde{I} \end{pmatrix} \begin{pmatrix} E \\ e \end{pmatrix} = \begin{pmatrix} -C_D^{-1}E - C_D^{-1}h^T \alpha c C_D^{-1}E + C_D^{-1}h^T \alpha e \\ \alpha c C_D^{-1}E - \alpha e \\ \tilde{I} \end{pmatrix} \quad (7.2)$$

$$, \alpha = (-c C_D^{-1} h^T + g)^{-1}$$

where $\tilde{I}$ has the rank $(n-r-1)$.

The matrix in (7.2) is equal to postmultiplying (7.1) by the following matrix,

$$\begin{pmatrix} \alpha c C_D^{-1}E - \alpha e \\ \tilde{I} \end{pmatrix} \quad (7.3)$$

The reduced Hessian before entering the new constraint is:

$$A = \begin{pmatrix} -h C_D^{-1} & \\ -E^T C_D^{-1} & \end{pmatrix} \nabla^2 f(x) \begin{pmatrix} -C_D^{-1} h^T, & -C_D^{-1} E \\ & I \end{pmatrix} \quad (7.4)$$

When a new constraint boundary is encountered, because of (7.3), the updated reduced Hessian in the $(n-r-1)$ subspace can be written as

$$(f^T, I)A \begin{pmatrix} f \\ \tilde{I} \end{pmatrix} \quad (7.5)$$

where $f = \alpha c C^{-1}E - ae$.

There are two cases for Newton's method but here we will just consider the case when the reduced Hessian is positive definite (and therefore has inverse).

Note that (7.5) is the upper left hand corner of

$$\begin{pmatrix} f^T & \tilde{I} \\ 1 & 0 \end{pmatrix} A \begin{pmatrix} f & 1 \\ \tilde{I} & 0 \end{pmatrix} \quad (7.6)$$

where 0 is a $1 \times (n-r-1)$ row vector of zeros.

A is positive definite and,

$$\begin{pmatrix} f & 1 \\ \tilde{I} & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & \tilde{I} \\ 1 & -f \end{pmatrix}.$$

Let

$$\begin{pmatrix} g_1 & g^T \\ g_2 & G_3 \end{pmatrix}$$

be the inverse of A in (7.4). Then the inverse of (7.6) is:

$$\begin{pmatrix} 0 & I \\ 1 & -f \end{pmatrix} \begin{pmatrix} g_1 & g_2 \\ g_2 & G_3 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ I & -f^T \end{pmatrix} = \begin{pmatrix} G_3 & , & g_2 - G_3 f^T \\ g_2^T - f G_3 & , & g_1 g_2^T f^T - f g_2 + f g_3 f^T \end{pmatrix} \quad (7.7)$$

and the desired inverse is:

$$(\hat{A})^+ = G_3 - (g_2 - G_3 f^T)(g_1 - 2f g_2 + f G_3 f^T)^{-1}(g_2 - G_3 f^T)^T \quad (7.8)$$

8. Subalgorithmic Strategies and Computations

To determine the **subspace** in which the minimization should take place we used the following strategy.

At the beginning of each iteration, the algorithm computes the estimates of the **Lagrange** multipliers. If those for the binding inequality constraints are all non-negative, then the motion at iteration k is to be made in the linear **subspace** defined by the currently binding constraints, otherwise the inequality constraint with the least multiplier estimate is dropped from the list of those required to be binding during the iteration. A descent move in the resulting **subspace** is then attempted.

While minimizing $f(x)$ along the search vector S^k defined by

$$S^k = -B(B^T B)^+ (B^T \nabla f),$$

if before $f(x)$ is minimized a constraint's boundary is encountered say at some point $x^{k+1} = x^k + \alpha^k S^k$, we do not terminate the iteration, but we use the anti-zig-zagging strategy suggested by McCormick [23].

In [12], using classical Newton's method, α^k was found such that $f(x^k + \alpha^k s^k)$ is minimized along s^k . The iterative formula to find α^k is:

$$\alpha^{j+1} = \alpha^j - [h''(\alpha)]^{-1} h'(\alpha^j), \quad j=1,2,\dots \quad (8.1)$$

where

$$h(\alpha) = f(x^k + \alpha s^k), \text{ and } f \in C^2.$$

To ensure the convergence of the sequence generated by (8.1) lower and upper bounds are imposed on α .

If at j^{th} iteration the current bounds are L^j and U^j such that $L^j < \alpha^j < U^j$, then α^j is accepted and the bounds are updated as follows:

if $h'(\alpha^j) < 0$ then $L^{j+1} = \max(\alpha^j, L^j)$, $U^{j+1} = U^j$
 and if $h'(\alpha^j) > 0$ then $U^{j+1} = \min(\alpha^j, U^j)$, $L^{j+1} = L^j$.

9. Statement of the Algorithms

In this section a step by step presentation of both projection and Elimination algorithms are given

Projection Algorithm

- Step 1: Find the set of binding constraints, and therefore the matrix C .
- Step 2: Calculate $(CC^T)^{-1}$. In higher iterations, i.e. $k > 1$, if a constraint is to be added or deleted from the set of binding constraints, update $(CC^T)^{-1}$ by (4.1) or (4.3).
- Step 3: Find the Lagrange multipliers $w^k = (CC^T)^{-1}C\nabla f$.
- Step 4: If all $w_1^k \geq 0$ for the inequality constraints, go to step 6, otherwise go to step 5.
- Step 5: Delete the inequality constraint with the least multiplier from the set of binding constraints and update $(CC^T)^{-1}$.
- Step 6: Calculate the estimate of the generalized inverse of the positive part of the projected Hessian.
- Step 7: Calculate the search vector $S^k = -A^+(P\nabla f(x))$, minimize $f(x)$ along S^k .
- Step 8: If $|f(x^{k+1}) - f(x^k)| < \epsilon$ stop. Otherwise go to step 1. ϵ is a predefined small positive number.

Elimination Algorithm

- Step 1: Determine the set of binding constraints, therefore the matrix C .
- Step 2: Find the set of dependent variables and the matrix C_D .
- Step 3: Find C_D^{-1} . At the beginning of iteration k where $k > 1$, update C_D^{-1} by (4.4) or (4.3) when a constraint becomes binding or a binding constraints leaves its boundary.
- Step 4: Calculate the Lagrange multipliers

$$w^k = (C_D^{-T}) \nabla_D f(x^k).$$
- Step 5: If $w^k \geq 0$ for all the inequality constraints, go to step 7, otherwise go to step 6.
- Step 6: Delete the inequality constraint with the least multiplier from the set of binding constraints. Update (C_D^{-1}) .
- Step 7: Calculate the generalized inverse of the estimated positive part of the reduced Hessian i.e., $A^+ = B^T \nabla^2 f(x) B$
- Step 8: Calculate $S^k = -B(A^+)(B^T \nabla f(x))$
- Step 9: Minimize $f(x)$ along S^k .
- Step 10: If $|f(x^{k+1}) - f(x^k)| < \epsilon$ stop. Otherwise go to step 1.

10. Comparison of the Algorithms and Conclusion

From (3.3) it can be concluded that two different search vectors generated with two different null space matrices in the same subspace are equal. Therefore the search vectors generated by the two algorithms will be equal at many points. The only difference between the two algorithms might occur from the Lagrange multiplier estimates and that is because these estimates are not the same for the two algorithms. If the multiplier associated with one of the binding constraints is negative in one algorithm and not in another, or if the least negative multiplier in both algorithms is not associated with the same binding constraint, the space in which the search vector is to be generated will become different and this difference might cause changes in the number of iterations.

Another area of comparison is computational effort and efficiency which are defined in [33] as the time of execution and the number of gradient or function calls.

In [12] some test problems were solved. Although the number of function, gradient, Hessian and first and second directional derivative computations were the same, there was still a slight difference in execution time which for large scale problems or for problems with large numbers of iterations might become considerable and make the elimination algorithm more efficient than the projection algorithm.

Also an important point to note is that the primary concern of this work has been in the development of the numerical aspects of solving linearly constrained nonlinear programming problems. The algorithms presented are not "globally convergent" algorithms in that they cannot be expected, in general, to produce sequences whose points of accumulation are local minimizers. They are "locally convergent" algorithms, that is, if started near an isolated local minimizer which satisfies certain regularity conditions, they should produce points which converge to that isolated local minimizer. The rate of convergence will be "at least quadratic." The regularity assumptions are the usual ones: second order sufficiency conditions, linear independence of the binding constraint gradients, and the strict positivity of the generalized **Lagrange** multipliers associated with the binding inequality constraints.

The proof of this assertion for a similar algorithm associated with the more difficult nonlinearly constrained problem is contained in **McCormick** [22].

In order to obtain global convergence, some modifications have been suggested in [12] to obtain a sequence of points in a region where the Hessian of the objective function is positive semi-definite with respect to vectors in the nullspace of the derivative matrix of the active constraints.

REFERENCES

- (1) Abadie, J. and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraint," in R. Fletcher (ed) "Optimization," Academic Press, London and New York, 1969, pp. 37-47.
- (2) Abadie, J. and Guigou, "Numerical Experiments with the GRG Method," in J. Abadie (ed) Integer and Nonlinear Programming, North Holland Publishing Co., Amsterdam, 1973, pp. 529-536.
- (3) Balinski, M.L. and E. Hellerman (eds.), Mathematical Programming Study 4: Computational Practice in Mathematical Programming, North Holland Publishing Co., 1975.
- (4) Berman, A. and A. Ben-Israel, "Linear Inequalities, Mathematical Programming and Matrix Theory," Mathematical Programming, 1, (1971) 291-300.
- (5) Ben-Israel, A. and T.N.E. Greville, Generalized Inverses: Theory and Application, John Wiley and Sons, Inc., New York, 1974.
- (6) Bracken, J. and G.P. McCormick, Selected Applications of Nonlinear Programming, John Wiley and Sons, Inc., New York, 1968.
- (7) Brown, K. and K. Hillstrom, M. Minkoff, L. Nazareth, J. Pool, B. Smith, "A Minpack Project, Process Report in the Development of Modularized Package of Algorithms for Optimization Problems," Argonne National Laboratory, Applied Mathematics Division, Technical Memorandum No. 225, 1974.
- (8) Emami, G., "Evaluating Strategies for Newton's Method Using a Numerical Stable Generalized Inverse Algorithm," D.Sc. Dissertation, The George Washington University, 1978.
- (9) Fletcher, R., "Minimizing a General Function Subject to Linear Constraints," a review paper presented at an International Conference on Optimization, University of Dundee, Scotland, 1971.
- (10) Fletcher, R., "An Algorithm for Solving Linearly Constrained Optimization Problems," Mathematical Programming, 2, (1972) 133-165.
- (11) Ghaemi, A. and G.P. McCormick, "Symbolic Factorable SUMT: What is it? How to Use It?" Technical Paper Serial T-402, 1979.
- (12) Ghotb, F., "Newton's Method for Linearly Constrained Optimization Problems", D.Sc. Dissertation, The George Washington University, 1980.
- (13) Gill, P.E. and W. Murray, "Newton Type Methods for Unconstrained and Linearly Constrained Optimization Problems," Mathematical Programming, 7, (1974) pp. 311-350.
- (14) Gill, P.E. and W. Murray (eds.), Numerical Methods for Unconstrained Optimization, Academic Press, London, 1974.
- (15) Gill, P.E. and W. Murray, "The Computation of Lagrange Multiplier Estimates for Constrained Minimization," Mathematical Programming, 17, (1979) 32-60.
- (16) Goldfarb, D., "Matrix Factorization in Optimization of Nonlinear Functions Subject to Linear Constraints," Mathematical Programming, 10, (1979) 1-31.

- (17) Goldfarb, D., "Extension of Davidon's Variable Matrix Method to Maximization Under Linear Inequality and Equality Constraints," *SIAM J., Applied Mathematics*, Vol. 17, No. 4, 1969.
- (18) Hock, N. and K. Schittkowski, "Test Example for Solution of Nonlinear Programming Problems, Part 1," 1979, Institute für Angewandte Mathematik und Statistik, Universität Würzburg, Am Hubland, D-37 Würzburg, West Germany.
- (19) Lenard, M.L., "A Computational Study of Active Set Strategy in Nonlinear Programming with Linear Constraints," *Mathematical Programming*, 16, (1979) 81-97.
- (20) Lidor, G., "Chemical Equilibrium Problem Treated by Geometric and Transcendental Programming," Technical Report, 1975, Dept. of Operations Research, Stanford University.
- (21) Marlow, W.H., Mathematics for Operations Research, John Wiley and Sons, Inc., New York, 1978.
- (22) McCormick, G.P., "An Arc Method for Nonlinear Programming," SIAM J. CONTROL, 13, (1975), 1194-1216.
- (23) McCormick, G.P., "Anti Zig-Zagging by Bending," *Management Science*, 15, No. 5, (1969) 315-320.
- (24) McCormick, G.P., "Formulas for Updating the Generalized Inverse of a Matrix Perturbed by a Dyad," Technical Paper T-333, Institute for Management Science and Engineering, The George Washington University, Washington, D.C., March 1976.
- (25) Meyer, Carl D., Jr., "Generalized Inversion of Modified Matrices," *SIAM J., Applied Mathematics*, Vol. 24, No. 3, (1973) 315-323.
- (26) Murtagh, B.A. and M.A. Saunders, "Large Scale Linearly Constrained Optimization," *Mathematical Programming*, 14, (1978) 41-72.
- (27) Murtagh, B.A. and R.W.H. Sargent, "A Constrained Minimization Method with Quadratic Convergence," in R. Fletcher (ed) Optimization, Academic Press, London and New York, 1969.
- (28) Penrose, R., "A Generalized Inverse for Matrices," *Proceedings of the Cambridge Philosophical Society* 51 (1955) 406-413.
- (29) Ritter, K., "A Variable Matrix Method for Linearly Constrained Minimization Problem," in O.L. Mangasarian, R.R. Meyer and S.M. Robinson (eds.), Nonlinear Programming, 3, Academic Press, New York, San Francisco, London, 1978.
- (30) Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming, I. Linear Constraints," *J. Soc. Indust. Applied Mathematics*, 8, (1960) 181-217.
- (31) Sandgren, E., "Utility of Nonlinear Programming Algorithms," Thesis, 1977. Purdue University.
- (32) Sargent, R.W.H. and B.A. Murtagh, "Projection Methods for Nonlinear Programming," *Mathematical Programming*, 4, (1973) 245-268.

- (33) Schittkowski, A., "A Numerical Comparison of 13 Nonlinear Programming Codes with Randomly Generated Test Problems," 1979. To appear in L.C.W. Dixon and G.P. Szego (eds, Numerical Optimization of Dynamic Systems, North Holland Publishing Co.
- (34) Shanno, D.F. and R.E. Marston, "Conjugate Gradient Methods for Linearly Constrained Nonlinear Programming,:" MIS Technical Report, 1979, Dept. of Management Information System, College of Business and Public Administration, University of Arizona.
- (35) Shanno, D.F., "A Modified Newton-Raphson Technique for Constrained Nonlinear Estimation Problems," Ph.D. Dissertation, 1966, Carnegie Institute of Technology, Pittsburgh, Pennsylvania.
- (36) Shayan, M.E., "A Methodology for Comparing Algorithms and a Method of Computing with Order Directional Derivatives Based on Factorable Programming," D.Sc. Dissertation, 1980, The George Washington University.
- (37) Stewart, G.W., Introduction to Matrix Computation, Academic Press, New York, San Francisco, London, 1973.
- (38) Waren, A.D. and L.S. Ladson, "The Status of Nonlinear Programming Software," *Operations Research*, 27, (1979) 431-456.
- (39) Wolfe, P., "Convergence Conditions for A Scant Method II: Some Corrections," *SIAM Review*, 13, (1971) 185-188.
- (40) Wolfe, P., "Methods of Nonlinear Programming," in R.L. Graves and P. Wolfe (eds), Recent Advances in Mathematical Programming, McGraw-Hill Book Co., 1963, pp. 67-86.
- (41) Zoutendijk, G. (ed). Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam and New York, 1960.

Swinburne Institute of TechnologyFaculty of BusinessLIST OF STAFF PAPERS PUBLISHED TO DATE

- No. 7 1981 'A Note on Customs Unions Theory: The Viner Controversy R.I.P.' by D.J. Thomas.
- No. 8 1981 'Disequilibrium and the Expectations-Augmented Phillips Curve' by Max Grant.
- No. 9 1981 'A View of Ideological Pressures in the Context of Managerial Power' by Max Brown.
- No. 10 1981 'Short Term Prediction of Student Numbers in the Victorian Secondary Education System' by Miles G. Nicholls.
- No. 11 1982 'The Legal Protection of Geographical Trade Names: Prognosis for a Case of Champagne' by Bruce Clarke.
- No. 12 1984 'Corporate Planning Practice in Major American and Australian Manufacturing Companies' by Noel Capon, Chris Christodoulou, John U. Farley and James Hulbert.
- No. 13 1984 'A Modified Markovian Direct Control Model in Fixed Time Incorporating a New Objective Function Specification' by Miles G. Nicholls.
- No. 14 1984 'Government Intervention in the Labour Market - A Case Study of the Referral and Placement Activity of the Commonwealth Employment Service in a Major Metropolitan Area' by John B. Wielgosz.
- No. 15 1984 'Big Business in the U.S. and Australia: A Comparative Study' by Noel Capon, Chris. Christodoulou, John U. Farley and James M. Hulbert.
- No. 16 1984 'Modelling the Demand for Tertiary Education - An Exploratory Analysis Based on a Modified Human-Capital Approach' by Miles G. Nicholls.
- No. 17 1984 'Formal Corporate Planning Practices of Major Australian Manufacturing Companies' by Chris Christodoulou.
- No. 18 1984 'The Australian Short Run Demand for Money Function. Further Theoretical Considerations and Empirical Evidence Using Bayesian Techniques' by Edgar J. Wilson.

- No. 19 1984 'Alternative Job Search and Job Finding Methods: Their Influence on Duration of Job Search and Job Satisfaction' by John B. Wielgosz and Susan Carpenter.
- No. 20 1985 'A Comprehensive Study of Strategic Planning in Australian Subsidiary and Non-Subsidiary Companies' by Chris Christodoulou and Peter T. Fitzroy.
- No. 21 1985 'Towards an Optimal Taxation Structure in Australia' by David J. Thomas.
- No. 22 1985 'A Suggested Theoretical Basis for the Interpretation of the Effects of Income on the Demand for Tertiary Education' by Miles G. Nicholls.
- No. 23 1985 'Austrian Economics and Australian Patents' by Bruce Oakman.
- No. 24 1985 'Ensuring a Future for your Organisation' by Chris Christodoulou.
- No. 25 1985 'The Long Search: A Pursuit of Organizational Understanding from the Perspective of "System" Thinkers' by Max Brown.
- No. 26 1985 'Managing the Introduction of New Technology' by John Newton.
- No. 27 1986 'Positive Economic Analysis and the Task of State Enterprise Efficiency and Control' by Patrick Xavier.
- No. 28 1986 'Profitability of Horizontal Takeovers in the Australian Industrial Equity Market: 1978 to 1982' by M.A. Johns and N.A. Sinclair.
- No. 29 1986 'A Comparative Examination of Subsidiary and Non-Subsidiary Strategies' by Chris Christodoulou.
- No. 30 1986 'Solving Linearly Constrained Nonlinear Programming Problems by Newton's Method' by Fatemeh Ghotb.