ITERATIVE LINEAR PROGRAMMING SOLUTION OF CONVEX PROGRAMS

Michael C. Ferris

Computer Sciences Technical Report #794

November 1988

Iterative Linear Programming Solution of Convex Programs

Michael C. Ferris*

Computer Sciences Department

University of Wisconsin

Madison, Wisconsin 53706

November 1988

Abstract. An iterative linear programming algorithm for the solution of the convex programming problem is proposed. The algorithm partially solves a sequence of linear programming subproblems whose solution is shown to converge quadratically, superlinearly or linearly to the solution of the convex program, depending on the accuracy to which the subproblems are solved. The given algorithm is related to inexact Newton methods for the nonlinear complementarity problem. Preliminary results for an implementation of the algorithm are given.

Key words. Iterative linear programming, complementarity problems, inexact Newton methods, finite termination

Abbreviated title. Iterative Linear Programming

*This material is based on research supported by National Science Foundation Grants DCR–8521228 and CCR–8723091 and Air Force Office of Scientific Research Grant AFOSR–86–0172

1 Introduction

We propose an iterative linear programming method to solve the nonlinear program

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \begin{array}{r} g(x) \leq 0 \\ x \geq 0 \end{array} \end{array} \tag{1}$$

where f and g are differentiable, convex functions. Problem (1) will be solved by partially solving the following sequence of linear programs

$$\begin{array}{ll} \text{minimize} & c(i,k)^T y \\ \text{subject to} & \begin{array}{r} M^i y + q^i \geq 0 \\ y \geq 0 \end{array} \end{array} \tag{2}$$

where i represents an outer loop iteration and k represents a finitely terminating inner iteration. We first describe the motivation and source of the quantities appearing in (2) and then give a precise description of the algorithm.

In fact the algorithm produces a sequence of iterates which converge to a Karush–Kuhn–Tucker point of (1). Under any of the standard constraint qualifications (see Mangasarian[7] for details), the Karush–Kuhn–Tucker conditions for (1) hold at any solution point. The Karush–Kuhn–Tucker conditions for this problem are

$$\begin{aligned} & v = \nabla_x L(x,u) = \nabla f(x) + u^T \nabla g(x) \geq 0 \\ & y = -\nabla_u L(x,u) = -g(x) \geq 0 \\ & x \geq 0, u \geq 0 \\ & x^T v = 0 \\ & u^T y = 0 \end{aligned}$$

where $L(x,u) = f(x) + u^T g(x)$ is the standard Lagrangian of the problem. By letting

$$z = \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+m} \quad \text{and} \quad F(z) = \begin{bmatrix} \nabla_x L(x,u) \\ -\nabla_u L(x,u) \end{bmatrix}$$

we produce the nonlinear complementarity problem, $NLCP(F)$, of finding $z \in \mathbb{R}^{n+m}$ with

$$F(z) \geq 0, \quad z \geq 0$$
$$z^T(F(z)) = 0$$

If we linearize $NLCP(F)$ about a current iterate z^i, the following linear complementarity problem, LCP(M^i, q^i), is obtained. Find $z \in \mathbb{R}^{n+m}$ such that

$$M^i z + q^i \geq 0, \quad z \geq 0$$
$$z^T(M^i z + q^i) = 0$$

The quantities appearing in this problem are precisely the same ones as appear in (2) and are given by

$$\begin{aligned} H^i &:= \nabla^2_{xx} L(z^i) \\ M^i &:= \begin{bmatrix} H^i & \nabla g(x^i)^T \\ -\nabla g(x^i) & 0 \end{bmatrix} \\ q^i &:= \begin{bmatrix} \nabla f(x^i) - H^i x^i \\ -g(x^i) + \nabla g(x^i) x^i \end{bmatrix} \end{aligned} \tag{3}$$

The algorithm we propose can be briefly described as follows. Linearize $NLCP(F)$ about the current iterate to form LCP(M^i, q^i). Solve this problem using a finitely convergent iterative linear programming technique (inner loop). Then check a termination criterion and repeat the process if the method has not converged (outer loop).

A precise description follows, where the iterates in the **outer loop** are indexed with a superscript i and in the **inner loop** by a superscript k.

2 Algorithm

Initialize: Given $\epsilon > 0$ and any point $z^0 = (x^0, u^0)$, set $i = 0$.

Outer loop: Given $z^i = (x^i, u^i)$, calculate the following quantities

$$F(z^i) := \begin{bmatrix} \nabla_x L(z^i) \\ -\nabla_u L(z^i) \end{bmatrix}, \quad H^i, \quad M^i, \text{ and } q^i$$

(the last three given by (3)). Find z^{i+1} by executing the **inner loop(i)**. If $\|z^{i+1} - z^i\| \leq \epsilon$, then stop. Otherwise set $i = i + 1$ and repeat the **outer loop**.

Inner loop(i)

Initialize: Find any basic feasible solution, say $z^{i,0}$ of the linear constraints in variable y

$$\begin{aligned} M^i y + q^i &\geq 0 \\ y &\geq 0 \end{aligned}$$

and set $k = 0$. (For example, use phase I of the simplex method.)

Iteration: Let

$$\theta_i(z) = z^T(M^i z + q^i)$$

Direction: Given $z^{i,k}$, use the primal simplex method to partially solve (2), with

$$c(i,k) := \nabla\theta_i(z^{i,k})$$

until

$$\nabla\theta_i(z^{i,k})^T y \leq \nabla\theta_i(z^{i,k})^T z^{i,k} - \theta_i(z^{i,k}) \tag{4}$$

Let $y^{i,k}$ be the basic feasible solution obtained satisfying (4). See [13] and the definition of M^i in (3) for justification of (4). Define

$$p^{i,k} := y^{i,k} - z^{i,k}$$

Steplength: Let $\lambda_{i,k}$ satisfy

$$\lambda_{i,k} := \arg\min_{0 \leq \lambda \leq 1} \theta_i(z^{i,k} + \lambda p^{i,k})$$

Set

$$z^{i,k+1} := z^{i,k} + \lambda_{i,k} p^{i,k}$$

Termination: If

$$\left\| h_i(z^{i,k+1}) \right\| \leq \eta_i \left\| \min(z^i, F(z^i)) \right\|$$

where $h_i(z)$ is defined by

$$h_i(z) := \min(z, M^i z + q^i)$$

and η_i is a sequence of scalars chosen to ensure convergence, then set $z^{i+1} := z^{i,k+1}$ and return to the **outer loop**. Otherwise, increase k by one and repeat the **iteration** of the **inner loop**(i).

(Note that the function h_i is vector–valued with "min" interpreted as a componentwise minimum.)

We now give a broad motivation for the above method, cite relevant convergence results and give computational results for a particular implementation of the above scheme. The algorithm comes from a consideration of two methods in the literature, namely

1. Pang's inexact Newton method for $NLCP(F)$[9].

2. Shiau's iterative linear programming algorithm for LCP(M, q)[13].

We now proceed to describe the essential elements of these methods.

3 Pang's inexact Newton method for $NLCP(F)$

Newton's method for solving $NLCP(F)$ generates a sequence $\{z^i\}$ in the following way (see Josephy[5] for details). Given z^i, define $z^{i+1} \in \mathbb{R}^{n+m}$ to be an exact solution of the linear complementarity subproblem:

$$F(z^i) + \nabla F(z^i)(z - z^i) \geq 0, \quad z \geq 0$$
$$z^T(F(z^i) + \nabla F(z^i)(z - z^i)) = 0$$

Under suitable assumptions, the sequence $\{z^i\}$ is well defined and converges locally and quadratically to a solution z^* of $NLCP(F)$. However, there are several well–known drawbacks of the method. The first is that the need to evaluate the Jacobian, $\nabla F(z^i)$, is computationally expensive and the second is the need to solve the subproblems exactly can be time consuming and sometimes impossible. In the first case, some headway has been made by using approximations to the Jacobian (see the work by Josephy[6] on quasi–Newton schemes, for example). We will not consider these methods here. For the second drawback, inexact Newton methods have been proposed, offering a trade–off between the accuracy of solving the subproblems and the amount of work needed to solve them. Pang[9] has described such a scheme and we outline his results here since they apply directly to the method that we propose. The method in its general form is given by:

Let z^i be the most recent iterate, then generate $z^{i+1} \in \mathrm{IR}^{n+m}$ as an approximate solution of

$$F(z^i) + \nabla F(z^i)(z - z^i) \geq 0, \quad z \geq 0$$
$$z^T(F(z^i) + \nabla F(z^i)(z - z^i)) = 0$$

The approximation must satisfy the following rule

$$\left\| h_i(z^{i+1}) \right\| \leq \eta_i \left\| \min(z^i, F(z^i)) \right\| \tag{5}$$

where $\eta_i > 0$ is some given scalar and

$$h_i(z) := \min(z, F(z^i) + \nabla F(z^i)(z - z^i))$$

It is clear that z^i is a solution of $NLCP(F)$ if and only if $\min(z^i, F(z^i)) = 0$. Thus, under suitable assumptions, the quantity $\|\min(z^i, F(z^i))\|$ measures how close z^i is to a solution of $NLCP(F)$. The main advantage of a rule such as (5) is the savings in the solution effort during the early iterations.

To state the convergence result we need to consider the notion of regularity.

Definition 1 (Robinson) *Let z^* be a solution of $NLCP(F)$. z^* is said to be* **regular** *if* *there exists a neighbourhood N^* of z^* and a scalar $\delta > 0$ such that, for all vectors y with*

$$\|y\| \leq \delta$$

there is a unique vector $z(y) \in \mathbb{R}^{n+m}$ in N^* solving the perturbed linear complementarity problem

$$F(z^*) - y + \nabla F(z^*)(z - z^*) \geq 0, \quad z \geq 0$$
$$z^T(F(z^*) - y + \nabla F(z^*)(z - z^*)) = 0$$

Moreover, $z(y)$ is Lipschitz continuous in y, that is, there is a constant $L > 0$ so that whenever $\|y\| < \delta$ and $\|y'\| < \delta$, we have

$$\|z(y) - z(y')\| \leq L\,\|y - y'\|$$

The above notion of regularity is due to Robinson[11] and is a generalization of a familiar condition in the case of solving systems of nonlinear equations. Indeed, if the $NLCP(F)$ is viewed as an extension of finding a zero of the mapping $F(z)$, then the regularity condition just defined can be thought of as generalizing the requirement that $\nabla F(\tilde{z})$ is nonsingular where $F(\tilde{z}) = 0$.

We now quote two lemmas, given by Pang[9], which explain some terms he mentions in his convergence proof of the inexact Newton method.

Lemma 2 (Pang) *Let F be Lipschitz continuous. Then there exists a constant $\mu > 0$ such that for all z and z'*

$$\|\min(z, F(z)) - \min(z', F(z'))\| \leq \mu\,\|z - z'\| \tag{6}$$

Condition (6) is used in the convergence proof. We include Lemma 2 to show that condition (6) holds under weak assumptions.

Lemma 3 (Pang) *Suppose that z^* is a regular solution of $NLCP(F)$ and that F is continuously differentiable in a neighbourhood of z^*. Then there exist a scalar $\delta > 0$, two* *neighbourhoods N_1 and N_2 of z^*, and a constant $L > 0$ such that whenever z is in N_1 and $\|y\| < \delta$, there is a unique vector $v(z,y)$ in N_2 satisfying*

$$v \geq 0, \quad w = F(z) - y + \nabla F(z)(v - z) \geq 0, \quad \text{and} \quad v^T(w) = 0$$

Moreover, if $\|y\| < \delta$ and $\|y'\| < \delta$, then

$$\|v(z,y) - v(z,y')\| \leq L\,\|y - y'\|$$

The following theorem, due to Pang[9], gives the convergence of the inexact Newton method (including rates of convergence) for the stopping rule (5).

Theorem 4 (Pang) *Suppose that z^* is a regular solution of $NLCP(F)$ and that F is continuously differentiable in a neighbourhood of z^*. Let $\mu > 0$ be such that condition (6) holds for all z and z' in a neighbourhood of z^*. Let L be the scalar as given in Lemma 3. Assume that*

$$\eta_i(1 + \max(L,1)\left\|I - \nabla F(z^i)\right\|) \leq \eta \min(1, 1/\mu) \quad \text{for all } i$$

for some $\eta < 1$. Then

- *there exists a neighbourhood of z^* so that if z^0 is chosen in it, the sequence, $\{z^i\}$, produced by the inexact Newton method under the approximate rule (5) is well defined and converges to z^**
- *if, in addition*

  $$\lim_{i\to\infty} \eta_i = 0$$

  then the convergence is superlinear, that is

  $$\lim_{i\to\infty} \frac{\|z^{i+1} - z^*\|}{\|z^i - z^*\|} = 0$$

- *if, in addition, ∇F is Lipschitz continuous in a neighbourhood of z^* and if, for all i sufficiently large*

$$\eta_i \leq \gamma \left\| \min(z^i, F(z^i)) \right\|$$

for some $\gamma > 0$, then the convergence is quadratic, that is

$$\left\| z^{i+1} - z^* \right\| \leq c \left\| z^i - z^* \right\|^2$$

for all i large enough and some $c > 0$.

Pang[9] also gives results concerning inexact quasi–Newton schemes, but since these do not relate to our implementation, we will not discuss them here. However, this does give an indication that quasi–Newton schemes could prove to be computationally effective for our method as well.

4 Shiau's iterative linear programming algorithm

We now describe a method for solving linear complementarity problems when the underlying matrix M is positive semidefinite. The method is due to Shiau[13]. Consider the following quadratic programming problem

$$\begin{array}{ll} \text{minimize} & \theta(z) := z^T(Mz + q) \\ \text{subject to} & \begin{array}{r} Mz + q \geq 0 \\ z \geq 0 \end{array} \end{array} \tag{7}$$

which is equivalent to solving LCP(M, q). In his thesis, Shiau[13] proposes the following algorithm for the solution of the problem.

Initialize: Find any basic feasible solution of the constraints, say z^0. (For example, by using phase one of the simplex method.)

Iteration: Repeat the following steps until z^k is complementary, that is

$$(z^k)^T(Mz^k + q) = 0$$

Direction: Given z^k, use the primal simplex method and pivot on the problem,

$$\begin{array}{ll} \text{minimize} & \nabla\theta(z^k)^T y \\ \text{subject to} & \begin{array}{r} My + q \geq 0 \\ y \geq 0 \end{array} \end{array}$$

until

$$\nabla\theta(z^k)^T y \leq \nabla\theta(z^k)^T z^k - \theta(z^k) \tag{8}$$

(It is shown in [13] that if no point y satisfies (8) then z^k is complementary.) Let y^k be the basic feasible solution obtained satisfying (8). Let the direction of search be

$$p^k := y^k - z^k$$

Steplength: Let λ_k be given by

$$\lambda_k := \arg \min_{0 \leq \lambda \leq 1} \theta(z^k + \lambda p^k)$$

Set

$$z^{k+1} := z^k + \lambda_k p^k$$

and increase k by one.

Shiau[13] gives a closed form for the above steplength as

$$\lambda_k = \begin{cases} \bar{\lambda} := -\nabla\theta(z^k)p^k / 2p^k M p^k & \text{if } 0 < \bar{\lambda} < 1 \\ 1 & \text{otherwise} \end{cases}$$

under the assumption that $\nabla\theta(z^k)p^k < 0$. He also proves that the algorithm terminates at an optimal solution in a finite number of steps. We cite Shiau's result now. Note that the hypothesis of the theorem is satisfied trivially under our assumption that M is positive semidefinite, but the given theorem is more general.

Theorem 5 (Shiau) *Assume that there exists a solution $\bar{z}$ of* LCP(M, q) *such that*

$$(y - \bar{z})^T M(y - \bar{z}) \geq 0 \quad \textit{for all feasible } y$$

Then there exists $N \geq 0$ such that

$$\theta(z^0) > \theta(z^1) > \cdots > \theta(z^N) = 0$$

where $\{z^k\}$ is generated by the above algorithm.

The algorithm we proposed at the beginning of this section is now seen to be precisely an inexact Newton method for the solution of $NLCP(F)$, using an adaptation of Shiau's method to (inexactly) solve LCP(M^i, q^i). These observations lead directly to the following result.

Theorem 6 *Under the assumptions of Theorem 4, the algorithm of Section 2 converges linearly, superlinearly or quadratically, depending on the choice of stopping criterion η_i.*

In passing, we make the following note. The subproblems generated by the inexact Newton method are in fact linear complementarity problems whose solution gives the Karush–Kuhn–Tucker conditions of the quadratic subproblems generated in the following iterative quadratic programming method:

Initialize: Given $z^0 = (x^0, u^0)$, set $i = 0$.

Iteration: Given $z^i = (x^i, u^i)$, find $z^{i+1} = (x^{i+1}, u^{i+1})$, a Karush–Kuhn–Tucker point for the following linearly constrained quadratic program

$$\begin{array}{ll} \text{minimize} & \nabla f(x^i)^T(x - x^i) + 1/2(x - x^i)^T G(z^i)(x - x^i) \\ \text{subject to} & \begin{array}{r} g(x^i) + \nabla g(x^i)(x - x^i) \leq 0 \\ x \geq 0 \end{array} \end{array} \tag{9}$$

Here $G(z^i)$ is some approximation to the Hessian of the Lagrangian of (1).

Termination: If some convergence criterion such as $\|x^{i+1} - x^i\| \leq \epsilon$ is satisfied, then stop. Otherwise, set $i = i + 1$ and repeat the **iteration.**

If we formulate the Karush–Kuhn–Tucker conditions of (9) as a linear complementarity problem, then it can be seen that at iteration i, the subproblem is equivalent to finding $z \in \mathbb{R}^{n+m}$ satisfying

$$\begin{aligned} M^i z + q^i \geq 0, \quad z \geq 0 \\ z^T(M^i z + q^i) = 0 \end{aligned}$$

Note that M^i and q^i are given by

$$M^i := \begin{bmatrix} G(z^i) & \nabla g(x^i)^T \\ -\nabla g(x^i) & 0 \end{bmatrix}$$

$$q^i := \begin{bmatrix} \nabla f(x^i) - G(z^i)x^i \\ -g(x^i) + \nabla g(x^i)x^i \end{bmatrix}$$

Thus if we take $G(z^i) = H^i$, the exact Hessian of the Lagrangian, the subproblems of this method can be viewed as the same subproblems as those given in our algorithm (see (3)). The idea of replacing the exact Hessian in these methods by an approximation has been the subject of much research. These are motivated by the iterative quadratic programming algorithm of Garcia Palomares and Mangasarian[2], Han[3] and Powell[10]. Second order information is used by this class of algorithms through various approximations to the Hessian of the Lagrangian function for the problem. For inexact quasi–Newton methods for $NLCP(F)$, the paper by Pang[9] gives a convergence proof under the assumption of regularity.

5 Computational Results

The method described above has been implemented using the exact Hessian formulation of the algorithm. The linear programming subproblems were solved using a reduced tableau implementation of the simplex method. The subproblems were started from an advanced basis, namely the one given by the previous iterate.

Problem	Size $m \times n$	Outer loops	Inner loops	Pivots	Basis pivots	CPU (secs)
Rosenbrock[12]	4×2	6	7	2	28	0.10
Himmelblau[4]	2×4	16	16	0	51	0.17
Wright[8]	3×5	6	6	0	52	0.15
Colville1[1]	10×5	3	4	4	25	0.28
Colville2[1] (Feas)	5×15	10	18	13	141	1.48
Colville2[1] (Infeas)	5×15	7	8	6	96	1.18

Table 1: ILP on standard test problems

Table 1 gives computational results when the algorithm was applied to several standard test problems of the literature. The column labelled 'basis pivots' gives the number of pivots that were required to regenerate the basis given by the previous outer loop iteration. Obviously, for a more sophisticated implementation of the simplex method, this can be accomplished more cheaply. All figures reported are totals given by adding the relevant figures per outer loop. The stopping rule (5) was used, with

$$\eta_i = 0.2 \left\| \min(z^i, F(z^i)) \right\|$$

The second set of test problems that were attempted can be described as follows. We look at the approximation of a given set of data $\{x_i, y_i\}$ by a function

$$f(x) = \alpha e^{\beta x}$$

where α and β are to be chosen in order to minimize some error in the approximation. We choose a 1–norm approximation, namely

$$\underset{\alpha,\beta}{\text{minimize}} \quad \left\| y_i - \alpha e^{\beta x_i} \right\|_1$$

and reformulate this problem as

$$\begin{array}{ll} \underset{\alpha,\beta,z}{\text{minimize}} & \sum_{i=1}^{n} z_i \\ \text{subject to} & -z_i \leq y_i - \alpha e^{\beta x_i} \leq z_i \\ & \quad \text{for } i = 1, \ldots, n \end{array}$$

We were then able to generate many nonlinear test problems of this form. The computational results of the proposed algorithm are given in Table 2 for a particular set of problems and are to be compared with those given by MINOS[8], reported in Table 3. Both methods were initialized at five different starting points which were generated randomly, under the condition that they lie in some ball about the (approximately known) solution point. The reader should compare the number of outer loops in Table 2 to the number of major iterations taken by MINOS. ILP does perform much better in this respect, although this may be due to the use of exact second order information. The pivot column of Table 2 should be compared with the number of minor iterations of Table 3. Again, ILP outperforms MINOS. It is difficult to compare function and constraint evaluations of the two methods since, on each inner loop, ILP calculates the function and constraints just once because it has the Hessian available in closed form. MINOS, however, uses an estimation scheme which, by its very nature, will be more expensive on the function and constraint calls. The CPU times are comparable, although ILP does take more time on the larger problems. This is due mainly to the inefficient scheme which was used to generate the advanced basis, whereby the previous known basic variables were pivoted into an artificial basis. It is conjectured that a more sophisticated implementation of the simplex method will enable these times to be dramatically reduced by constructing the new basis more efficiently. Furthermore, we hope to implement such a scheme and incorporate this into a quasi–Newton method to estimate the Hessian.

It is hoped that further research can be carried out on this class of methods, particularly with regard to schemes involving approximations of the Hessian of the Lagrangian.

Problem size	Outer loops	Inner loops	Pivots	Basis pivots	CPU (secs)
3	7.6	9.6	9.6	63.8	0.37
4	11.2	14.0	14.4	130.4	0.83
5	8.4	11.8	18.2	110.6	0.88
6	8.6	12.8	22.0	133.4	1.25
7	8.2	12.2	27.8	137.0	1.59
8	6.8	9.8	24.3	122.5	1.62
9	7.6	11.4	35.0	157.2	2.45
10	9.6	15.0	46.8	232.8	4.07
11	8.5	15.5	46.8	223.0	5.07
12	9.2	17.6	64.0	266.0	6.37
13	8.6	14.2	42.6	256.8	6.30
14	7.8	14.2	59.8	242.2	7.12
15	9.4	16.4	76.4	321.2	10.22
16	7.4	12.2	57.4	254.0	8.77
17	7.6	13.8	77.0	285.4	12.13
18	7.0	11.6	80.2	205.4	15.50
19	9.0	15.6	90.4	380.8	20.01
20	7.8	11.2	79.0	289.8	16.81

Table 2: ILP on approximation test problems

Problem size	Major iterations	Minor iterations	Function evaluations	Constraint evaluations	CPU (secs)
3	14.4	46.6	118.4	118.5	1.32
4	13.0	41.2	99.0	99.0	1.37
5	11.3	35.3	83.8	83.8	1.27
6	13.6	39.8	89.8	90.2	1.61
7	10.6	52.2	103.8	104.0	1.80
8	12.6	50.0	103.0	103.5	1.98
9	18.2	52.6	120.0	120.4	2.55
10	16.2	51.8	110.8	110.8	2.54
11	23.4	63.4	135.0	137.0	3.70
12	13.0	60.6	126.4	128.0	3.02
13	18.6	77.8	176.6	176.8	4.52
14	11.6	75.8	163.2	164.2	4.59
15	32.0	113.6	451.4	453.2	10.78
16	10.0	58.8	105.2	106.0	3.44
17	15.8	94.8	188.2	189.4	5.65
18	37.6	174.6	961.2	964.4	26.34
19	17.2	117.0	229.2	229.8	7.25
20	13.0	103.0	168.2	170.1	6.52

Table 3: MINOS on approximation test problems

Acknowledgement

I should like to thank Professor O. Mangasarian for useful discussions during the preparation of this paper.

References

[1] A.R. Colville. *A Comparative Study on Nonlinear Programming Codes.* Technical Report 320–2949, IBM New York Scientific Center, June 1968.

[2] U.M. Garcia Palomares and O.L. Mangasarian. Superlinearly convergent Quasi–Newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.

[3] S.–P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.

[4] D.M. Himmelblau. *Applied Nonlinear Programming.* McGraw–Hill, New York, 1972.

[5] N.H. Josephy. *Newton's Method for Generalized Equations.* Technical Summary Report 1965, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, 1979.

[6] N.H. Josephy. *Quasi–Newton Methods for Generalized Equations.* Technical Summary Report 1966, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, 1979.

[7] O.L. Mangasarian. *Nonlinear Programming.* McGraw–Hill, New York, 1969.

[8] B.A. Murtagh and M.A. Saunders. *MINOS 5.0 User's Guide.* Technical Report SOL 83.20, Stanford University, December 1983.

[9] J.S. Pang. Inexact Newton methods for the nonlinear complementarity problem. *Mathematical Programming*, 36(1):54–71, 1986.

[10] M.J.D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 3*, pages 27–63, Academic Press, London, 1978.

[11] S.M. Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5:43–62, 1980.

[12] H.H. Rosenbrock. Automatic method for finding the greatest or least value of a function. *Computer Journal*, 3:175–184, 1960.

[13] T.-H. Shiau. *Iterative Linear Programming for Linear Complementarity and Related Problems.* Technical Report 507, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin 53706, August 1983. PhD thesis.