An NE/SQP Method for the Bounded Nonlinear Complementarity Problem [1] [2] [3]

Steven A. Gabriel
Mathematics and Computer Science Divsion
Argonne National Laboratory

Preprint MCS-P508-0495
May 30, 1995

Abstract

NE/SQP is a recent algorithm that has proven quite effective for solving the pure and mixed forms of the nonlinear complementarity problem (NCP). NE/SQP is robust in the sense that its direction-finding subproblems are always solvable; in addition, the convergence rate of this method is Q-quadratic. In this paper we consider a generalized version of NE/SQP proposed by Pang and Qi, that is suitable for the *bounded* NCP. We extend their work by demonstrating a stronger convergence result and then test a proposed method on several numerical problems.

Key Words: Nonlinear complementarity problem, mathematical programming, sequential quadratic programming.

1 Introduction

In a recent paper [18] Pang and Qi presented a general algorithm to solve various mathematical programs that can be formulated as systems of nonsmooth equations. Various convergence results were shown for several general nonsmooth formulations. One specific nonsmooth system considered was the *upper bounded nonlinear complementarity problem* (UBNCP).

The UBNCP differs from the standard version in that the variables, normally bounded below by zero, are additionally bounded above. Upper bounds on the variables in an NCP are actually a frequent feature in many nonlinear programs and in energy and economic applications [22] [3] [13].

When the variables are bounded from above and below, it suffices to consider only the upper bounded version of the NCP. This is because if the vector of variables, lower bounds and upper bounds are, respectively, $x, l, u \in R^n$, with $l < u$, then without loss of generality we can replace x by the translated vector $\tilde{x} = x - l$ and apply the usual lower bound of zero and a new upper bound $\tilde{u} = u - l$.

In [18] Pang and Qi established convergence results for an NE/SQP method for the UBNCP; see [9, 10, 11, 17]) for a discussion of the NE/SQP method and its variants. Their proposed algorithm

[1] Dr. Gabriel is a postdoctoral researcher in the Mathematics and Computer Science Division of Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, email: gabriel@mcs.anl.gov.

[2] The research for Dr. Gabriel was conducted while a member of the Mathematical Sciences Department at The Johns Hopkins University in Baltimore, Maryland, and the Mathematics and Computer Science Division at Argonne National Laboratory (ANL). At ANL this work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

[3] The author thanks Prof. Philip Gill of UC San Diego for his help with the software QPOPT and Prof. Jong-Shi Pang of The Johns Hopkins University for many useful discussions concerning this work especially in its original form in the author's doctoral dissertation.

handled the upper bounds explicitly by incorporating them into a direction-finding subproblem; we refer to this approach as the upper-bounded NE/SQP method (UBNE/SQP).

This approach has several immediate advantages over using the bounds as additional constraints and solving the associated augmented NCP. First, the NCP function may not even be defined for variables outside these bounds, so maintaining the iterates within them easily permits computations involving such functions. Second, the problem size can be greatly reduced by considering these bounds as part of the subproblem only; see below for a more detailed explanation.

In this paper, we extend the convergence results for the upper-bounded NE/SQP method by showing that under a local Lipschitzian assumption this algorithm is actually *Q-quadratically* convergent; previously, it was known to be Q-superlinearly convergent. We conclude by comparing the UBNE/SQP with the NE/SQP algorithm as applied to an associated augmented NCP that handles the upper bounds as explicit constraints.

2 The Upper Bounded Nonlinear Complementarity Problem

Let $a \in R^n$ be a given positive vector and $f : R^n_+ \to R$ be a once continuously differentiable function. Then, the upper-bounded nonlinear complementarity problem is to find a vector pair $(x, y) \in R^n \times R^n$ such that the following conditions hold:

$$\begin{aligned} u = f(x) + y \geq 0, \quad x \geq 0, \quad u^T x = 0, \\ v = a - x \geq 0, \quad y \geq 0, \quad v^T y = 0. \end{aligned} \tag{1}$$

Notice that if we let $z^T = (x^T, y^T)$ and let $w(x,y)^T = (u(x,y)^T, v(x,y)^T)$ (where $u := u(x,y), v := v(x,y)$), then we have the equivalent NCP of order $2n$ given by

$$w(z) \geq 0, \quad z \geq 0, \quad w(z)^T z = 0. \tag{2}$$

In principle, any method that solves a general NCP could be applied to this augmented system. However, it is desirable, especially for large-scale applications, to take advantage of the specific structure of the problem given in (1) and develop an algorithm that maintains the variables within their bounds and works on a problem of the original size.

Another reason to avoid using the augumented system (2) concerns the Jacobian matrix $\nabla w(z)$. It is not hard to see that

$$\nabla w(z) = \begin{pmatrix} \nabla f(x) & I \\ -I & 0 \end{pmatrix}.$$

A condition needed in various NCP algorithms such as NE/SQP is that certain principal submatrices of $\nabla w(z)$ be nonsingular at a limit point of iterates; see for example the *b-regularity* condition for NE/SQP. By using this augmented form, we introduce a zero block in the lower right corner of $\nabla w(z)$ and potentially violate these nonsingularity conditions more often as a result.

We first define the *minimum sum map* $H : R^n_+ \to R^n$ as

$$H(x) = \min(x, f(x)_+) + \min(a - x, f(x)_-). \tag{3}$$

This function H is locally Lipschitzian and is intimately related to the UBNCP, as is shown in the next lemma.

Lemma 2.1 *Consider any $a \in R^n$ with $a > 0$ and let the function H be defined as in (3). Then x^* solves the UBNCP if and only if $H(x^*) = 0$ and $x^* \in [0, a]$.*

Proof. It suffices to observe that a vector $x^* \in [0, a]$ is a solution to the UBNCP if and only if the following system holds:

1. $x_i^* = 0 \Rightarrow f_i(x) \geq 0$,
2. $0 < x_i^* < a_i \Rightarrow f_i(x) = 0$, and
3. $x_i^* = a_i \Rightarrow f_i(x) \leq 0$. $\square$

Remark: Note that if such an x^* exists as given above, then the associated vector y^* is uniquely determined as follows: $y_i^* = 0$ when $x_i^* \in [0, a_i)$ and $y_i^* = -f_i(x^*) \geq 0$, when $x_i^* = a_i$. For this reason, it is sufficient to say that x^*, by itself, solves the UBNCP.

Motivated by the nonsmooth function H shown above, in the next section we present several general nonsmooth concepts relevant to the UBNE/SQP method to be described below.

3 Nonsmooth Analysis

Consider a locally Lipschitzian function $G : R^n \to R^n$. By Rademacher's theorem [6], G is almost everywhere F-differentiable. Denoting the set of points where G is F-differentiable by D_G, we can define for any $x \in R^n$, the *generalized subdifferential* of G at x in the sense of Clarke by

$$\partial G(x) = \text{conv}\{\lim \nabla G(x^j) : x^j \to x, x^j \in D_G\}.$$

It is known that this is a nonempty set that is both convex and compact.

Employing this concept, we can define the notion of a semismooth function. In what follows, we use the notation $y \to_h x$ to mean $y \to x, y \neq x$ and $\frac{y-x}{\|y-x\|} \to \frac{h}{\|h\|}, h \neq 0$.

Definition 3.2 *Let $x, h \in R^n$, with $h \neq 0$. Then the function $G : R^n \to R^n$ is said to be* semismooth *at x if G is locally Lipschitzian there and $\forall h \in R^n, h \neq 0$,*

$$\lim_{y \to_h x} \{Vh : V \in \partial G(y)\} \tag{4}$$

exists.

It is interestng to note that semismoothness at x implies directional differentiability there with $G'(x, h)$ equal to the above limit for all $h \neq 0$; see Proposition 2.1 in [20]. Also, the class of semismooth functions is quite large in the sense that it includes the smooth functions, convex functions, and piecewise smooth functions. Additionally, the sums, differences, products, and composites of semismooth functions are semismooth. This fact allows us to conclude that the function H defined by (3) is semismooth.

Another generalization of gradients for nonsmooth functions involves the notion of a *subgradient* [4]. For a concave function $\psi : D \subseteq R^n \to R$, a subgradient at a point $x \in D$ is defined by a vector $b(x)$ that satisfies

$$\psi(z) - \psi(x) - b(x)^T(z - x) \leq 0, \quad \forall z \in D.$$

Now, suppose that $z := x + th \in D$, for some $t > 0$ and $\forall h \in R^n$. Then we obtain

$$\frac{\psi(x+th) - \psi(x) - tb(x)^T h}{t} \leq 0, \quad \forall h \in R^n.$$

The next concept, that of *upper subgradients*, extends the above idea for functions ψ that are not necessarily concave.

Definition 3.3 *A function $\psi : R^n \to R$ is said to be* upper subdifferentiable *on a set $D \subseteq R^n$ if there exists a function $b : D \to R^n$ such that for all $x \in D$ and $h \in R^n$*

$$\limsup_{y \to x, y \in D, t \downarrow 0} \frac{\psi(y+th) - \psi(y) - tb(y)^T h}{t} \leq 0. \tag{5}$$

We call b an upper subgradient function *of ψ on D and $b(x)$ an* upper subgradient *of ψ at the point x.*

Remark: Note that when D is an open set and ψ is continuously differentiable on D, then ψ is upper subdifferentiable there with $\nabla\psi(x)$ serving as an upper subgradient. This fact will be heavily used in the convergence analysis that follows. One immediate consequence of the definition of upper subdifferentiability is that upper subgradients are additive in a certain sense. This is borne out in the next lemma.

Lemma 3.4 *Let $f, g : E \subseteq R^n \to R$ be given functions and let b_f and b_g be upper subgradient functions on a common domain D for f and g, respectively. Then $b_f + b_g$ is an upper subgradient function on D for $(f+g) := f + g$.*

Proof. For all vectors $h \in R^n$, we have

$$\begin{aligned}
& \limsup\nolimits_{y \to x, y \in D, t \downarrow 0} \{(f+g)(y+th) - (f+g)(y) - t(b_f(y) + b_g(y))^T h\}/t \\
\leq\ & \limsup\nolimits_{y \to x, y \in D, t \downarrow 0} \{f(y+th) - f(y) - tb_f(y)^T h\}/t \ + \\
& \limsup\nolimits_{y \to x, y \in D, t \downarrow 0} \{g(y+th) - g(y) - tb_g(y)^T h\}/t \\
\leq\ & 0,
\end{aligned}$$

where the last inequality follows from the definition of upper subdifferentiability for b_f and b_g. □

It is interesting to note that the upper subgradient is related to several known directional derivatives. For example, if ψ is locally Lipshitzian, then putting y equal to x in (5) yields $b(x)^T h \geq \psi^D(x,h)$ for all $h \in R^n$, where $\psi^D(x,h)$ is the upper Dini directional derivative defined by

$$\psi^D(x,h) = \limsup_{t \downarrow 0} \frac{\psi(x+th) - \psi(x)}{t}.$$

The upper subgradient function is also related to the Clarke and Michel-Penot directional derivatives [14] of a locally Lipschitzian function at a point x. The Clarke directional derivative at x in the direction h is given by

$$\psi^o(x,h) = \limsup_{y \to x, t \downarrow 0} \frac{\psi(y+th) - \psi(y)}{t},$$

and the Michel-Penot directional derivative is defined by

$$\psi^{\diamond}(x,h) = \sup_{k \in R^n} \limsup_{t \downarrow 0} \frac{\psi(x+th+tk) - \psi(x+tk)}{t}.$$

Each of these directional derivatives induces an associated subdifferential given respectively by

$$\partial\psi(x) = \{u \in R^n : u^T h \leq \psi^{o}(x,h), \forall h \in R^n\}$$

and

$$\partial^{\diamond}\psi(x) = \{u \in R^n : u^T h \leq \psi^{\diamond}(x,h), \forall h \in R^n\}.$$

The following result (Proposition 4 in [18]) summarizes the relationship between an upper subgradient and these various other notions.

Proposition 3.5 *Suppose that $\psi : R^n \to R$ is locally Lipschitzian on the set $D \subseteq R^n$. If ψ is upper subdifferentiable on D with upper subgradient function $b(\cdot)$, then for each $x \in D$,*

$$b(x) \in \partial^{\diamond}\psi(x) \subseteq \partial\psi(x).$$

Hence, for any $h \in R^n$,

$$\psi^{D}(x,h) \leq b(x)^T h \leq \psi^{\diamond}(x,h) \leq \psi^{o}(x,h).$$

The next lemma calculates upper subgradients for the special case of a function that is the minimum of two continuously differentiable functions. This is a crucial lemma for the convergence analysis that follows.

Lemma 3.6 *Let $r, s : E \subseteq R^n \to R$, be two continuously differentiable functions with $\psi(x) := min(r(x), s(x))$. Also, let the following sets be defined:*

$$\begin{aligned} T_r &:= \{x \in R^n : r(x) < s(x)\}, \quad T_s := \{x \in R^n : r(x) > s(x)\}, and \\ T_e &:= \{x \in R^n : r(x) = s(x)\}. \end{aligned}$$

Then the following conditions hold:

(a) *$\nabla r(x)$ is an upper subgradient of ψ at each point $x \in T_r$,*

(b) *$\nabla s(x)$ is an upper subgradient of ψ at each point $x \in T_s$, and*

(c) *if T_e is partitioned into arbitrary subsets T'_e and T''_e, then $\nabla r(x)$ and $\nabla s(x)$ are upper subgradients of ψ at each point $x \in T'_e$ and T''_e, respectively.*

Proof. Conditions (a) and (b) follow from the earlier fact about upper subgradients of continuously differentiable functions on open sets. Now let $x \in T_e(x)$ and let the function $b(y)$ be defined by

$$b(y) = \begin{cases} \nabla r(y), \text{ if } y \in T_r \cup T'_e \\ \nabla s(y), \text{ if } y \in T_s \cup T''_e. \end{cases}$$

Then we have the following:

$$\begin{aligned} & \limsup_{y \to x, t \downarrow 0} \tfrac{\psi(y+th) - \psi(y) - tb(y)^T h}{t} \\ = \; & \limsup_{y \to x, t \downarrow 0} \tfrac{\min(r(y+th), s(y+th)) - \min(r(y), s(y)) - tb(y)^T h}{t} \end{aligned}$$

$$\leq \begin{cases} \limsup_{y\to x, t\downarrow 0} \frac{r(y+th)-r(y)-t\nabla r(y)^T h}{t}, \text{ if } y \in T_r \cup T'_e \text{ and} \\ \limsup_{y\to x, t\downarrow 0} \frac{s(y+th)-s(y)-t\nabla s(y)^T h}{t}, \text{ if } y \in T_s \cup T''_e. \end{cases}$$

So we see that

$$\begin{aligned} & \limsup_{y\to x, t\downarrow 0} \frac{\psi(y+th)-\psi(y)-tb(y)^T h}{t} \\ \leq\ & \max(\limsup_{y\to x, t\downarrow 0} \frac{r(y+th)-r(y)-t\nabla r(y)^T h}{t}, \\ & \ \limsup_{y\to x, t\downarrow 0} \frac{s(y+th)-s(y)-t\nabla s(y)^T h}{t}) \\ \leq\ & 0. \end{aligned}$$

The last inequality is valid because r and s are continuously differentiable functions on the open set $\{z \in R^n : z = y + th, y \to x, t \downarrow 0, h \text{ fixed}\}$. $\square$

This lemma indicates, for the given function, how the notion of an upper subgradient generalizes that of a gradient when differentiability at a point may not exist. In what follows, we will use this last result for the *mininum sum map H* defined in (3). First we describe the NE/SQP method for the UBNCP as presented in [18].

4 The Upper Bounded NE/SQP Method

In view of the result from Lemma 2.1, we can solve the UBNCP by finding a constrained zero of the function H. If we define the norm function $\theta(x)$ equal to $\frac{1}{2}H(x)^T H(x)$, we see that for a given $a \in R^n_{++}$, a solution x^* of UBNCP corresponds to a minimizer of the following mathematical program with $\theta(x^*) = 0$:

$$\begin{aligned} \text{minimize }_x \quad & \theta(x) \\ \text{subject to} \quad & x \in X := [0, a]. \end{aligned} \tag{6}$$

Like the standard NE/SQP method, the upper bounded version is also an iterative algorithm. The latter method attempts to find an $\bar{x} \in R^n$ satisfying the constraints of (6) where $\theta(\bar{x}) = 0$. At each iteration, there is a specific quadratic program to solve whose solution will, in general, be a descent direction for the merit function θ.

Similar to the case of the standard NE/SQP, the functions $\phi : X \times R^n \to R_+$ and $z : X \times R^n \to R_+$ are respectively, the subproblem objective function and the forcing function and are defined as follows:

$$\phi(x, d) = \frac{1}{2}\sum_{i=1}^{n}(|H_i|(x) + b_i(x)^T d)^2$$

and

$$z(x, d) = \frac{1}{2}\sum_{i=1}^{n}(b_i(x)^T d)^2,$$

where $b_i(x)$ is an upper subgradient of the function $|H_i|(x) := |H_i(x)|$ at x on $\{x : |H_i(x)| \neq 0\} \cap X$ (i.e., the nonzero set of $|H_i|$), and it is an arbitrary vector in the Michel-Penot subdifferential if $H_i(x) = 0$. At the kth iteration of this method, given an iterate $x^k \in X$, the resulting quadratic program subproblem (QP$_k$) is to solve

$$\begin{aligned} &\text{minimize}_{\ d} \quad \phi(x^k, d) \\ &\text{subject to} \quad x^k + d \in X. \end{aligned} \tag{7}$$

Remark: For the function H defined above, we note that $|H_i|(x) = H_i(x)$ for all $i = 1, \cdots, n$ when $x \in X$.

In the convergence analysis of this method, it will be important to know specific values for the vector $b_i(x)$ as described above. The next result indicates that on the zero set of H_i (i.e., $\{x : H_i(x) = 0\}$), the zero vector can be used for $b_i(x)$. The proof of this result relies on the fact that the Michel-Penot directional derivative for the function $|H_i|$, that is, $|H_i|^\diamond(x, d)$, majorizes the usual directional derivative $|H_i|'(x, d)$; this follows from the inequalities presented in Proposition 3.5.

Lemma 4.7 *Consider any $a \in R^n$ with $a > 0$ and let the function H be defined as in (3). If $H_i(x) = 0$, then $0 \in \partial^\diamond |H_i|(x)$.*

Proof. First note that the directional derivative $|H_i|'(x, h)$ exists and is nonnegative for all $h \in R^n$. Its existence is guaranteed because H was shown to be semismooth. Thus H_i and $|H_i|$ are also semismooth, hence directionally differentiable. As for the nonnegativity, suppose for the sake of contradiction that this is not the case. Then for some $h \in R^n$, $|H_i|'(x, h) < 0$. Hence, there exists a $\bar{t} > 0$ such that for all $t \in [0, \bar{t})$ $|H_i|(x + th) < |H_i|(x) = 0$, which is a contradiction, since the function $|H_i|(\cdot)$ is always nonnegative. Consequently, with h any vector in R^n, we have $|H_i|^\diamond(x, h) \geq |H_i|'(x, h) \geq 0$, which shows that $0 \in \partial^\diamond |H_i|(x)$. □

On the nonzero set of H_i, there is some flexibility in choosing the vector $b_i(x)$ in the subproblem (7). However, for the convergence analysis that follows, it is important to specify particular values for $b_i(x)$ when $x \in X$ and $H_i(x) \neq 0$. In the next set of lemmas we describe explicit values for upper subgradients $b_i(x)$ of $|H_i|(x)$ on this set. We first consider the possible relationships between the pairs of arguments $(x_i, f_i(x)_+)$ and $(a_i - x_i, f_i(x)_-)$ for $i = 1, 2, \cdots, n$.

Lemma 4.8 *Let $f : R^n_+ \to R^n$ be a given function that is continuously differentiable. Consider the vectors $x, a \in R^n$ with $a > 0$ and $x \in [0, a]$. Then the set of indices $\{1, \cdots, n\}$ can be partitioned into the following seven sets:*

$$\begin{aligned}
&S_1(x) := \{i : (x_i < f_i(x)_+\}, \\
&S_2(x) := \{i : a_i - x_i < f_i(x)_-\}, \\
&S_3(x) := \{i : (x_i \geq f_i(x)_+, a_i - x_i = f_i(x)_-, f_i(x) < 0)\}, \\
&S_4(x) := \{i : (x_i = f_i(x)_+, a_i - x_i \geq f_i(x)_-, f_i(x) > 0)\}, \\
&S_5(x) := \{i : (x_i \geq f_i(x)_+, a_i - x_i > f_i(x)_-, f_i(x) < 0)\}, \\
&S_6(x) := \{i : (x_i > f_i(x)_+, a_i - x_i \geq f_i(x)_-, f_i(x) > 0)\}, \\
&S_7(x) := \{i : f_i(x) = 0\}.
\end{aligned}$$

Proof. The result follows from the fact that $a_i \geq x_i \geq 0$ must be maintained for all indices $i = 1, \cdots, n$. □

Remark: For use with the functions ϕ and z defined above, the seven index sets can be partitioned into four sets as follows:

$$S_1(x), S_2(x), S_7(x), \text{ and } T(x) = (S_1(x) \cup S_2(x) \cup S_7(x))^c.$$

However, part of the convergence analysis of the UBNE/SQP method requires that the indices in the set $T(x)$ be analyzed according to their membership in $S_3(x), S_4(x), S_5(x)$, or $S_6(x)$. For this reason we maintain the seven index sets in what follows.

Using these seveb index sets, we can analyze the possible values of the function $|H_i|(\cdot)$. The next lemma establishes, for each of these seven index sets, what values this function can take on at points that are arbitrarily close to each other. This is directly relevant in the analysis of the Q-quadratic convergence rate which appears below.

Lemma 4.9 *Consider the function H as defined in (3). Let $z, x, a \in R^n$, with $a > 0$ and $z, x \in [0, a]$. Then, if z is sufficiently close to x, the following values of $|H_i|(x)$ are valid:*

$$\begin{array}{llll}
(a) & |H_i|(z) = z_i, & |H_i|(x) = x_i & i \in S_1(z); \\
(b) & |H_i|(z) = a_i - z_i, & |H_i|(x) = a_i - x_i & i \in S_2(z); \\
(c) & |H_i|(z) = a_i - z_i = -f_i(z), & |H_i|(x) = min(a_i - x_i, -f_i(x)) & i \in S_3(z); \\
(d) & |H_i|(z) = z_i = f_i(z), & |H_i|(x) = min(x_i, f_i(x)) & i \in S_4(z); \\
(e) & |H_i|(z) = -f_i(z), & |H_i|(x) = -f_i(x) & i \in S_5(z); \\
(f) & |H_i|(z) = f_i(z), & |H_i|(x) = f_i(x) & i \in S_6(z); \\
(g) & |H_i|(z) = 0, & |H_i|(x) = 0 & i \in S_7(z).
\end{array}$$

Proof. Since $f(\cdot)$ and $|H_i|(\cdot)$ are continuous functions and z is sufficiently close to x, the above statements follow by examining the relationships described in each of the index sets in question. $\square$

We can also specify possible upper subgradient values for $|H_i|$ on its nonzero set via the index sets listed above.

Lemma 4.10 *Consider the function H defined as in (3). Let $z, a \in R^n$, with $a > 0$ and $z \in [0, a]$. Then, if $i \in \{1, \cdots, n\} - S_7(z)$, the following values of $b_i(z)$ are upper subgradients of the function $|H_i|$ at the point z; (e_i is the ith standard basis vector):*

$$b_i(z) = \begin{cases} e_i & i \in S_1(z) \\ -e_i & i \in S_2(z) \\ -e_i, -\nabla f_i(z) & i \in S_3(z) \\ e_i, \nabla f_i(z) & i \in S_4(z) \\ -\nabla f_i(z) & i \in S_5(z) \\ \nabla f_i(z) & i \in S_6(z). \end{cases}$$

Proof. The result follows from noting the assertions given in Lemmas 3.4 and 3.6 as applied to the index sets in question. $\square$

Remark: In the UBNE/SQP method, we take $b_i(x) = 0$ if $i \in S_7(x)$ (for which $H_i(x) = 0$) and $b_i(x)$ as given above otherwise. This approach is consistent with the requirements on b_i, imposed in the definition of ϕ. Note that even when $H_i(x) = 0$ but $i \notin S_7(x)$ the values of $b_i(x)$ listed above, which are used in the UBNE/SQP method, are valid, since by Proposition 3.5 upper subgradients are necessarily elements of the Michel-Penot subdifferential.

Before presenting the UBNE/SQP method, we introduce a version of s-regularity for the UB-NCP. The reader will note the similarity with the definition in [17] applied to the standard NCP.

Definition 4.11 *Let a be a given vector in R^n_{++} and $X := [0, a]$. Then the* cone of feasible directions *of X at a point $x \in X$, denoted $\mathcal{F}_X(x)$, is defined by $\{d : x + \varepsilon d \in X, \ \ \forall$ sufficiently small $\varepsilon > 0\}$.*

A point $x \in X$ is said to be s-regular *if for every $b(x) = (b_i(x)) \in \Pi_{i=1}^n \partial |H_i|(x)$, there exists a direction $d \in \mathcal{F}_X(x)$ such that for each i where $H_i(x) \neq 0$, we have*

$$|H_i|(x) + b_i(x)^T d \leq 0.$$

We now present the UBNE/SQP method.

Algorithm 4.12 Upper-Bounded NE/SQP

Step 0. (Initialization) Having a vector $a \in R^n_{++}$, select $\rho, \sigma \in (0, 1)$, and an arbitrary vector $x^0 \in X := [0, a]$. Set $k = 0$.

Step 1. (Direction generation) Given $x^k \in X$, solve the (convex) quadratic program (QP$_k$) using the choices for $b_i(x^k)$ as explained above; let d^k be an arbitrary optimal solution obtained. If $\phi(x^k, d^k) = \theta(x^k)$, terminate the algorithm; otherwise, continue.

Step 2. (Step length determination) Let m_k be the smallest nonnegative integer m such that

$$\theta(x^k + \rho^m d^k) - \theta(x^k) \leq -\sigma \rho^m z(x^k, d^k); \tag{8}$$

set $x^{k+1} = x^k + \rho^{m_k} d^k$.

Step 3. (Termination check) If x^{k+1} satisfies a prescribed termination rule, stop. Otherwise, return to Step 1 with k replaced by $k + 1$.

Remark: In Step 1, when the condition $\phi(x^k, d^k) = \theta(x^k)$ is met and x^k is s-regular, then x^k solves the UBNCP; see Proposition 10, part (c), of [18].

We next describe the major convergence result for the UBNE/SQP method presented in [18]. First, however, we need to introduce a new regularity concept.

Definition 4.13 *Given $a \in R^N_{++}$, we call the vector $z \in [0, a]$* gb-regular *(for generalized b-reguar) if for all index sets α such that*

$$S_5(z) \cup S_6(z) \subseteq \alpha \subseteq S_3(z) \cup S_4(z) \cup S_5(z) \cup S_6(z),$$

the principal submatrix $\nabla_\alpha f_\alpha$ is nonsingular.

The reader will notice the similarity between the notion of gb-regularity and that of b-regularity described in [17]. We have the following convergence result.

Theorem 4.14 *[18] Consider any $a \in R^n$ with $a > 0$ and let $H : R^n_+ \to R^n$ be defined as in (3). Suppose that x^* is a limit point of a sequence $\{x^k\}$ produced by the UBNE/SQP method with associated directions $\{d^k\}$. If x^* is both s-regular and gb-regular, then $H(x^*) = 0$ and $x^* \in [0, a]$, that is, x^* solves the UBNCP. Moreover,*

(i) *there exists an integer $K > 0$ such that $\forall k \geq K$, the step length $\tau_k = 1$; hence $x^{k+1} = x^k + d^k$; and*

(ii) *the sequence $\{x^k\}$ converges to x^** Q-superlinearly*; in other words,*

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

It remains to show that under a suitable assumption on the function f, the UBNE/SQP method is actually Q-quadratically convergent.

Theorem 4.15 *Consider any $a \in R^n$ with $a > 0$ and let $H : R^n_+ \to R^n$ be defined as in (3). Let x^* be a limit point of the sequence of iterates $\{x^k\}$ generated by the UBNE/SQP method with associated directions $\{d^k\}$. If x^* is both s-regular and gb-regular, then x^* is a solution to the UBNCP. Furthermore, if $f : R^n_+ \to R^n$ is Lipschitzian in a neighborhood around x^*, then the rate of convergence is* Q-quadratic*; in other words, $\exists$ a constant $c' > 0$ such that*

$$\limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < c'.$$

Proof. The fact that x^* is a solution was shown in Theorem 4.14, so we proceed to the rate of convergence result. From the proof of Theorem 4 in [18], we know that there exists a $c > 0$ such that the following useful inequality is valid for all k large enough:

$$c\|x^k + d^k - x^*\| \leq \| \ |H|(x^*) - |H|(x^k) - b(x^k)(x^* - x^k)\|. \tag{9}$$

Here $b(x^k)$ is an $n \times n$ matrix with the ith row being the transpose of $b_i(x^k)$, which is an upper subgradient of $|H_i|(x^k)$ at x^k if $H_i(x^k) \neq 0$ and a vector in $\partial^\diamond|H_i|(x^k)$ otherwise. We now show that the right-hand side of (9) is bounded above by $\|f(x^*) - f(x^k) + \nabla f(x^k)(x^* - x^k)\|$.

In what follows, k is taken large enough so that x^k is sufficiently close to x^*. We consider the following expression:

$$| \ |H_i|(x^*) - |H_i|(x^k) - b_i(x^k)^T(x^* - x^k)|. \tag{10}$$

If $i \in S_7(x^k)$, then by Lemma 4.9, $H_i(x^k) = H_i(x^*) = 0$. Since Lemma 4.7 indicates that $b_i(x^k) = 0$ is valid here, we see that (10) is equal to zero. If $i \in S_1(x^k) \cup S_2(x^k)$, then using the suggested values in Lemmas 4.9 and 4.10 we see that the same expression is also zero. Also, for $i \in S_5(x^k) \cup S_6(x^k)$, (10) has the value

$$|f_i(x^*) - f_i(x^k) - \nabla f_i(x^k)^T(x^* - x^k)|. \tag{11}$$

For the remaining index sets $S_3(x^k)$ and $S_4(x^k)$, based on the result from Lemma 4.9, there are two cases to consider for each index set, depending on which argument is the minimum. In all cases however, we get a term equal to 0 or of the form given in (11). So we see that there exist constants $c_1, c_2 > 0$ such that the following is valid:

$$\begin{aligned}\|x^k + d^k - x^*\| &\le c^{-1}\| \ |H|(x^*) - |H|(x^k) - b(x^k)(x^* - x^k)\| \\ &\le c_1\| \ |H|(x^*) - |H|(x^k) - b(x^k)(x^* - x^k)\|_1 \\ &\le c_1\|f(x^*) - f(x^k) - \nabla f(x^k)(x^* - x^k)\|_1 \\ &\le c_2\|f(x^*) - f(x^k) - \nabla f(x^k)(x^* - x^k)\|.\end{aligned}$$

In view of the fact that f is assumed Lipschiztian in a neighborhood around x^*, and by part (i) in Theorem 4.14, we see that the desired result follows. □

5 Numerical Experiments

5.1 Test Problems

A Nash-Cournot Production Problem [NC]

This numerical problem first appeared in [16], it concerned a simplified production model under the Nash-Cournot equilibrium framework. The defining function $f : R^n_+ \to R^n$ is of the form

$$f_i(q) = c_i'(q_i) - p(\sum_{j=1}^n q_j) - q_i p'(\sum_{j=1}^n q_j), \quad i = 1, \ldots, n,$$

where

$$c_i(q_i) = \alpha_i q_i + \frac{\beta_i}{1+\beta_i} L_i^{-1/\beta_i} q_i^{1+1/\beta_i}, \quad p(Q) = 5000^{1/\gamma} Q^{-1/\gamma}$$

with $Q = \sum_{j=1}^n q_j$. The data α_i, L_i, β_i, and γ are positive scalars. Notice that the function c_i is not twice differentiable at $q_i = 0$ if $\beta_i > 1$. By means of a straightforward calculation, it can be shown that $\nabla f(q)$ is a P-matrix for any positive vector q, as long as $\gamma > 1$. From the relevant literature, the solutions that have been obtained had all variables positive.

Two Optimization Problems [HS-100,HS-113]

We have selected two minimization problems (#100 and #113) from [12]. Both problems are of the form

$$\min \psi(x) : g(x) \le 0,$$

where $\psi : R^n \to R$ and $g : R^n \to R^m$. For problems #100 and #113, respectively, $(n, m) = (7, 4)$ and $(10, 8)$. The Karush–Kuhn–Tucker (KKT) optimality conditions for these mathematical programs give rise to the function $f : R^n \times R^m_+ \to R^{n+m}$ given by

$$f(x, \lambda) = \begin{pmatrix} \nabla\psi(x) + \sum_{i=1}^m \lambda_i \nabla g_i(x) \\ -g(x) \end{pmatrix},$$

which is a mixed NCP in the variables x and λ. For problem #113, we have also imposed the constraints that $x \ge 0$ since the reported optimal solution has all positive values; the resulting KKT conditions produce a pure NCP.

A Spatial Price Equilibrium Problem [SPE]

Ever since Samuelson's suggestion that this equilibrium problem is amenable for analysis by mathematical programming [23], numerous formulations and solution procedures based on this approach have been proposed and studied. To summarize the discussion, we give the NCP formulation of a multicommodity version of the model that appears in [8]. The defining function of the NCP is given by

$$f(x,\pi) = \begin{pmatrix} \pi_i^k + c_a^k(x) - \pi_j^k & : & a = \text{arc joining nodes } i \text{ to } j \\ S_l^k(\pi) - D_l^k(\pi) - Q_l^k(x) & : & l \in N \end{pmatrix},$$

where N denotes the nodes of the network, $k = 1, \ldots, K$ denotes the commodities, and $Q_l^k(x)$ is the net supply of commodity k at node l:

$$Q_l^k(x) = \sum_{a \in T(l)} x_a^k - \sum_{a \in H(l)} x_a^k,$$

where $T(l)$ $(H(l))$ is the set of all arcs whose tail (head, resp.) is the node l. The transportation cost functions $c_a^k(x)$ and the supply and demand functions $S_l^k(\pi)$ and $D_l^k(\pi)$ can take on different forms. In particular, the following polynomial functions have been used:

$$\begin{aligned} c_a^k(x) &= \Gamma_a^k + \Omega_a^k (x_a^k)^4 + \textstyle\sum_{j \neq k} \delta_a^{kj} x_a^j \\ S_l^k(\pi) &= H_l^k + J_l^k (\pi_l^k)^2 + \textstyle\sum_{j \neq l} u_{lj}^k \pi_j^k \\ D_l^k(\pi) &= E_l^k - G_l^k (\pi_l^k)^2 + \textstyle\sum_{j \neq l} w_{lj}^k \pi_j^k. \end{aligned}$$

Note that these are asymmetric functions in general.

A Traffic Equilibrium Problem [TE]

The general traffic equilibrium problem was formulated as a variational inequality problem in [24]. Its NCP formulation involves the function f defined by (see [2, 1])

$$f(h,u) = \begin{pmatrix} \Delta^T c(\Delta h) - \Gamma u \\ \Gamma^T h - D(u) \end{pmatrix}$$

where h, u are respectively, the path flows and origin-destination shortest times vectors, Δ is the (arc, path) incidence matrix of a network, and Γ is the (path, OD-pair) incidence matrix, that is,

$$\Delta_{ap} = \begin{cases} 1 & \text{if arc } a \text{ is incident to path } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$\Gamma_{pk} = \begin{cases} 1 & \text{if path } p \text{ joins OD-pair } k \\ 0 & \text{otherwise.} \end{cases}$$

There are various forms for the arc cost function $c(x)$ where $x = \Delta h$ is the vector of arc flows, and for the travel demand function $D(u)$. The special case of a constant demand function is important in its own right. That of a separable arc cost function is also noteworthy; this is the case where $c_a(x)$ depends only on x_a; for example,

$$c_a(x) = \alpha_a + \beta_a x_a^4.$$

This cost function is commonly used in practical traffic analysis; see [1]. An example of an asymmetric demand function is that derived from a logit model:

$$D_{ij}(u) = d_{ij}^0 \frac{r_{ij} e^{-u_{ij}}}{\sum_{k \neq i} r_{ik} e^{-u_{ik}}}, \quad i \neq j$$

with $r_{ij} = e^{\gamma_{ij}}$ being positive constants; see [1].

It should be pointed out that when the NCP(f) with f as given above is solved by a Newton-type method (such as NE/SQP), a complete knowledge of all the paths of the traffic network is required in advance. In order to avoid such an (often prohibitive) enumeration of the paths, some methods for solving this problem have employed the idea of path-generation embedded within a Newton scheme; see [19] [5]. In the latter reference, the authors develop an NE/SQP-based method that solves the traffic equilibrium problem by generating paths as the algorithm proceeds, thereby making it attractive for large networks.

5.2 Results

In what follows, we report the results of various numerical tests on several NCP problems. All the relevant programs were written in FORTRAN using double-precision accuracy. We have used the software QPOPT [21] to solve each quadratic program subproblem and performed our computations on a SPARCstation 5.

For each test problem, we have tried two starting points (x^a and x^b) and two sets of upper bound vectors a. The upper bounds are somewhat natural for the problems being considered. For example, these upper bounds could represent limits on primal or dual variables (HS-100, HS-113), or bounds on network flows as a result of capacity limitations, etc. The first method of choosing the upper bound vector a was to take $a_i = 1000$ for all i. [4] This corresponded to the case where the upper bounds in a typical application would be far from tight.

In the second method for selecting a, we first took the solution vector x^* and then applied the following logic:

$$a_i = |x_i^*|, \text{ if } i > .2 * n, \text{ or } a_i < 1.d - 5 \text{ then set } a_i = 20.$$

The point of this approach was to set a certain fraction of the upper bounds (the first 20%) so that at a solution, the corresponding upper bounds would be tight. The remaining variables would have bounds that were not too far off (i.e., 20 higher than x_i^*). The exception to this approach was if x_i^* was very small (i.e. $|x_i^*| < 1.d - 5$), the associated search direction might not be useful since the upper and lower bounds would be essentially zero; hence we forced those indices to have an upper bound of 20.

Here are the starting points for each of the problems.

NC We set $x_i^a = 10$ for all i and $x_i^b = 1$ for all i.

[4] The exception being for HS-100, where $-100 \leq x_i \leq 100$ for $i = 1, \ldots, 7$, since these variables were unconstrained and optimal values were much smaller than 100.

HS-100 We set x_i^a, $i = 1, \ldots, 7$ was chosen as in [12]; $x_8^a, \ldots, x_{11}^a = 1$. The second starting point was chosen so that $x_i^b = 1$ for all i.

HS-113 We took $x_1^a, \ldots, x_{10}^a$ as given in [12] and $x_{11}^a, \ldots, x_{18}^a$ were set to a value of 1. For the second initial vector we took $x_i^b = 1$ for all i.

SPE The vector x^a was set to all ones and x^b was taken to be all zeroes.

TE The vector x^a matched starting point a in [7]. The second vector x^b was calcuated as follows: for $x_1, \ldots, x_{38}$ (the path flow vector) we used the starting arc flows as suggested in [1] and solved for the starting path flows using the arc-path incidence matrix Δ. It is interesting to note that this approach yielded some negative components; the algorithms did not have a problem with such a point, however. [5]

Note that for all tests, we initiated the variable y (as described above) at zero. In the tables below, the designation "UBNE/SQP" refers to using the UBNE/SQP method whereas "Aug. NE/SQP" refers to using the standard NE/SQP on the augmented system (of size $2n$) as described above. The stopping criteria were as follows for each i:

$$\begin{array}{ll} \text{for UBNE/SQP} & |\tilde{H}_i(x)| = |\min(x_i, f_i(x))| \leq 1.d-6 \\ \text{for Aug. NE/SQP} & |H_i(x)| \leq 1.d-6. \end{array}$$

Also, the quantity $\theta(x)_i$ refers, respectively to $\frac{1}{2}H_i(x)^2$ or $\frac{1}{2}\tilde{H}_i(x)^2$ if UBNE/SQP or the augmented system approach was used. Lastly, the column "improv." measures the CPU improvement in using UBNE/SQP over NE/SQP as applied to the augmented system. The results are summarized in Tables 1-4.

Table 1
upper bound =method #1
starting point= group a

Problem	size	$\theta(x^0)$	$\theta(x^*)$	# of iters.	CPU (secs.)	improv.
NC–UBNE/SQP	10	3,492.2	6.1D-18	6	0.10	
NC–Aug. NE/SQP	20	3,492.2	6.1D-18	6	0.20	50 %
HS-100–UBNE/SQP	11	730.0	1.2D-16	12	0.17	
HS-100–Aug. NE/SQP	22	730.0	8.2D-26	9	0.39	56 %
HS-113–UBNE/SQP	18	6,934.5	2.3D-16	28	2.16	
HS-113–Aug. NE/SQP	36	6,934.5	8.7D-16	23	4.20	49 %
SPE–UBNE/SQP	42	7,538.3	4.2D-16	12	7.66	
SPE-Aug. NE/SQP	84	7,538.3	6.3D-16	9	20.32	62 %
TE–UBNE/SQP	50	1,300.9	2.4D-15	7	8.21	
TE–Aug. NE/SQP	100	1,300.9	1.0D-24	8	30.47	73 %

[5] For HS-113, method #2 and starting point x^a, UBNE/SQP was not able to eventually satisfy the bounds. This is because the starting point was not within these bounds. The resulting solution was just slightly off, and so we adjusted a by setting $a_2 = 3$ (instead of approximately 2.36368).

Table 2
upper bound =method #1
starting point= group b

Problem	size	$\theta(x^0)$	$\theta(x^*)$	# of iters.	CPU (secs.)	improv.
NC–UBNE/SQP	10	321,987.9	1.7D-15	9	0.12	
NC–Aug. NE/SQP	20	321,987.9	1.7D-15	9	0.30	60 %
HS-100–UBNE/SQP	11	8,918.5	1.5D-14	11	0.17	
HS-100–Aug. NE/SQP	22	8,918.5	9.5D-17	10	0.43	60 %
HS-113–UBNE/SQP	18	200,859.6	2.5D-14	16	1.24	
HS-113–Aug. NE/SQP	36	200,859.6	1.1D-23	18	3.58	65 %
SPE–UBNE/SQP	42	7,625.0	6.3D-14	9	4.51	
SPE-Aug. NE/SQP	84	7,625.0	9.7D-26	9	20.22	78 %
TE–UBNE/SQP	50	1,359.2	1.2D-33	9	9.96	
TE–Aug. NE/SQP	100	1,359.2	3.1D-20	10	37.23	73 %

Table 3
upper bound =method #2
starting point= group a

Problem	size	$\theta(x^0)$	$\theta(x^*)$	# of iters.	CPU (secs.)	improv.
NC–UBNE/SQP	10	3,427.1	2.8D-19	6	0.08	
NC–Aug. NE/SQP	20	3,492.2	2.7D-15	6	0.19	58 %
HS-100–UBNE/SQP	11	730.0	2.5D-15	11	0.18	
HS-100–Aug. NE/SQP	22	730.0	8.2D-26	9	0.41	56 %
HS-113–UBNE/SQP	18	701.6	3.8D-13	34	1.74	
HS-113–Aug. NE/SQP	36	6,934.7	4.0D-18	21	4.16	58 %
SPE–UBNE/SQP	42	4,163.9	2.1D-16	8	4.15	
SPE-Aug. NE/SQP	84	7,538.3	4.7D-16	9	21.08	80 %
TE–UBNE/SQP	50	1083.7	6.3D-17	16	14.71	
TE–Aug. NE/SQP	100	1300.9	1.9D-14	7	29.96	51 %

Table 4
upper bound =method #2
starting point= group b

Problem	size	$\theta(x^0)$	$\theta(x^*)$	# of iters.	CPU (secs.)	improv.
NC–UBNE/SQP	10	22,918.4	1.9D-18	5	0.05	
NC–Aug. NE/SQP	20	321,987.9	1.7D-15	9	0.31	84 %
HS-100–UBNE/SQP	11	8,918.5	1.9D-16	14	0.24	
HS-100–Aug. NE/SQP	22	8,918.5	9.5D-17	10	0.44	45 %
HS-113–UBNE/SQP	18	1,733.9	8.8D-20	20	1.12	
HS-113–Aug. NE/SQP	36	200,859.6	3.3D-22	17	3.49	68 %
SPE–UBNE/SQP	42	4,331.5	2.6D-16	8	3.96	
SPE-Aug. NE/SQP	84	7,625.0	1.1D-17	9	20.82	81 %
TE–UBNE/SQP	50	1,293.3	3.4D-16	10	10.0	
TE–Aug. NE/SQP	100	1,359.2	1.9D-19	10	37.54	73 %

6 Conclusions

In this paper, we have considered algorithms for solving the upper-bounded nonlinear complementarity problem. We have extended the results concerning the recent NE/SQP-type algorithm proposed by Pang and Qi by showing that it is actually Q-quadratically convergent under a suitable Lipschitzian assumption. Based on our test examples, we have additionally shown that the proposed algorithm, which handles the upper bounds in the subproblem and consequently solves small problems, can be much more efficient than the method that treats these upper bounds via an augmented NCP.

References

[1] Aashtiani, H. Z., *The Multi-Modal Traffic Assignment Problem*, Ph.D. thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1979.

[2] Aashtiani, H. Z. and Magnanti, T. L., *Equilibria on a Congested Transportation Network*, SIAM Journal on Algebraic and Discrete Methods, 2, pp. 213–226, 1981.

[3] Ahn, B. H. *Computation of Market Equilibria for Policy Analysis: The Project Independence Evaluation System (PIES) Approach*, Garland Publishing, Inc., New York, 1979.

[4] Bazaraa, M. S. and Shetty, C. M., *Nonlinear Programming Theory and Algorithms*, John Wiley & Sons, New York, 1979.

[5] Bernstein, D. H., and Gabriel, S. A., unpublished information, Mathematics and Computer Science Division, Argonne National Laboratory, May 1995.

[6] Clarke, F. H., *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983.

[7] De Luca, T., Facchinei, F., and Kanzow, C., *A Semismooth Equation Approach to the Solution of Nonlinear Complementarity Problems*, manuscript, University of Hamburg, Institute of Applied Mathematics, Bundesstrasse 55, D-20146, Hamburg, Germany, 1995.

[8] Friesz, T. L., Tobin, R. L., Smith, T. E., and Harker, P. T., *A Nonlinear Complementarity Formulation and Solution Procedure for the General Derived Demand Network Equilibrium Problem*, Journal of Regional Science, 23, pp. 337–359, 1983.

[9] Gabriel, S. A. *Algorithms for the Nonlinear Complementarity Problem: The NE/SQP Method and Extensions*, Ph.D. thesis, The Johns Hopkins University, 1992.

[10] Gabriel, S. A. and Pang, J. S., *An Inexact NE/SQP Method for Solving the Nonlinear Complementarity Problem*, Computational Optimization and Applications, 1, pp. 67–91, 1992.

[11] Gabriel, S. A. and Pang, J. S., *A Trust Region Method for Constrained Nonsmooth Equations*, in *Large-Scale Optimization: State of the Art*, edited by W. W. Hager, D. W. Hearn. and P. M. Pardalos, Kluwer Academic Publishers, pp. 159–186, Boston, 1994.

[12] Hock, W., and Schittkowski, K., *Test Examples for Nonlinear Programming Codes*, Springer-Verlag, Berlin, 1981.

[13] Hogan, W. W., *Energy Policy Models for Project Independence*, Computers and Operations Research, 2, pp. 251–271, 1975.

[14] Michel, P., and Penot, J. P., *Calcul Sous-différential pour des Fonctions Lipschitziennes et Non Lipschitziennes*, Comptes Rendues de l'Academie des Sciences Paris, 298, pp. 269–272, 1984.

[15] Mifflin, R., *Semismooth and Semiconvex Functions in Constrained Optimization*, SIAM Journal on Control and Optimization, 15, pp. 959–972, 1977.

[16] Murphy, F. H., Sherali, H. D., and Soyster, A. L., *A Mathematical Programming Approach for Determining Oligopolistic Market Equilibrium*, Mathematical Programming, 24, pp. 92–106, 1982.

[17] Pang, J. S. and Gabriel, S. A., *NE/SQP: A Robust Algorithm for the Nonlinear Complementarity Problem*, Mathematical Programming, 60, pp. 295–337, 1993.

[18] Pang, J. S., and Qi, L., *Nonsmooth Equations: Motivation and Algorithms*, SIAM Journal on Optimization, 3, pp.443–465, 1993.

[19] Pang, J. S., and Yu, C. S., *Linearized Simplicial Decomposition Methods for Computing Traffic Equilibria on Networks*, Networks, 14, pp. 427–438, 1984.

[20] Qi, L. and Sun, J., *A Nonsmooth Version of Newton's Method*, Mathematical Programming, 58, pp. 353–367, 1993.

[21] Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide for QPOPT 1.0: A FORTRAN Package for Quadratic Programming*, Department of Mathematics, University of California, San Diego, 1995.

[22] Rutherford, T. F., *The GAMS Callable Program Library for Variational and Complementarity Problems*, Department of Economics, University of Western Ontario, London, Ontario, 1991.

[23] Samuelson, P. A., *Spatial Price Equilibrium and Linear Programming*, American Economic Review, 42, pp. 283–303, 1952.

[24] Smith, M. J. , *The Existence, Uniqueness and Stability of Traffic Equilibria*, Transportation Research, 13B, pp. 295–304, 1979.