Journal of the Operations Research
Society of Japan
Vol. 37, No. 4, December 1994

OPTIMIZATION BASED GLOBALLY CONVERGENT METHODS FOR THE NONLINEAR COMPLEMENTARITY PROBLEM

Kouichi Taji Masao Fukushima
Nara Institute of Science and Technology

(Received June 14, 1993; Revised January 28, 1994)

Abstract The nonlinear complementarity problem has been used to study and formulate various equilibrium problems including the traffic equilibrium problem, the spatial equilibrium problem and the Nash equilibrium problem. To solve the nonlinear complementarity problem, various iterative methods such as projection methods, linearized methods and Newton method have been proposed and their convergence results have been established. In this paper, we propose globally convergent methods based on differentiable optimization formulation of the problem. The methods are applications of a recently proposed method for solving variational inequality problems, but they take full advantage of the special structure of nonlinear complementarity problem. We establish global convergence of the proposed methods, which is a refinement of the results obtained for variational inequality counterparts. Some computational experience indicates that the proposed methods are practically efficient.

1. Introduction

We consider the nonlinear complementarity problem, which is to find a vector $x \in R^n$ such that

$$x \geq 0, F(x) \geq 0 \text{ and } x^T F(x) = 0, \tag{1}$$

where $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T$ is a given continuously differentiable mapping from R^n into itself and ^{T} denotes transposition. This problem has been used to formulate and study various equilibrium problems including the traffic equilibrium problem, the spatial economic equilibrium problem and Nash equilibrium problem [1, 6, 13, 15, 18].

To solve the nonlinear complementarity problem (1), various iterative algorithms, such as fixed point algorithms, projection methods, nonlinear Jacobi method, successive over-relaxation methods and Newton method, have been proposed [8, 10, 16]. Many of these methods are generalizations of classical methods for systems of nonlinear equations and their convergence results have been studied extensively [10, 16].

Assuming the monotonicity of mapping F, Fukushima [7] has recently proposed a differentiable optimization formulation for variational inequality problem and proposed a decent algorithm to solve variational inequality problem. Based on this optimization formulation, Taji et al. [17] proposed a modification of Newton method for solving the variational inequality problem, and proved that, under the strong monotonicity assumption, the method is globally and quadratically convergent.

In this paper we apply the methods of Fukushima [7] and Taji et al. [17] to the nonlinear complementarity problem. We show that those methods can take full advantage of the special structure of problem (1), thereby yielding new algorithms for solving strongly monotone complementarity problems. We establish global convergence of the proposed methods, which is a refinement of the results obtained for the variational inequality counterparts in several respects. In this paper we show that the compactness assumption made in [7] can be removed for the strongly monotone complementarity problem. Moreover, some computational results shows that the proposed methods are practically efficient for solving monotone complementarity problems, though the convergence of the proposed methods is theoretically proved only under the strong monotonicity assumption.

2. Equivalent optimization problem

In this section, we introduce a merit function for the nonlinear complementarity problem (1) and present some of its properties.

Choose positive parameters $\delta_i > 0, i = 1, 2, \ldots, n$ and define the function $f : R^n \to R$ by

$$f(x) = \sum_{i=1}^{n} \frac{1}{2\delta_i} \left\{ F_i(x)^2 - (\max(0, F_i(x) - \delta_i x_i))^2 \right\}. \tag{2}$$

This function is a special case of the function originally introduced by Fukushima [7] for variational inequality problem. Though, some of its properties can be derived from the results of [7], we give here simple and direct proofs for these properties, which utilize the special structure of problem (1).

For convenience, we define

$$f_i(x) = \frac{1}{2\delta_i}\{F_i(x)^2 - (\max(0, F_i(x) - \delta_i x_i))^2\}, \tag{3}$$

hence f is written as $f(x) = \sum_{i=1}^{n} f_i(x)$. We denote D as a diagonal matrix such that $D = diag(\delta_1, \delta_2, \ldots, \delta_n)$, and e_i as the i-th unit vector such that

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} < i.$$

We also denote

$$M(x) = \max(0, x - D^{-1}F(x)), \tag{4}$$

where max operator is taken component-wise, i.e.,

$$M_i(x) = \max(0, x_i - \delta_i^{-1} F_i(x)).$$

Using this notation, we have the following result.

Lemma 1 *Let the mapping $M : R^n \to R^n_+$ be defined by (4). Then x^* solves (1) if and only if $x^* = M(x^*)$.*

Proof. Suppose that x^* is a solution to (1). Then either

$$\begin{cases} x_i^* = 0 \\ F_i(x^*) \geq 0 \end{cases} \text{or} \begin{cases} x_i^* \geq 0 \\ F_i(x^*) = 0 \end{cases}$$

holds for all $i = 1, 2, \ldots, n$. Since $\delta_i > 0$, we have

$$x_i^* = 0 \Longrightarrow M_i(x^*) = \max(0, -\delta_i^{-1} F_i(x)) = 0,$$

and

$$F_i(x^*) = 0 \Longrightarrow M_i(x^*) = \max(0, x_i^*) = x_i^*.$$

Thus $M_i(x^*) = x_i^*$ for all i.

On the other hand, suppose that x^* satisfies $x^* = M(x^*)$. Then, for all i, either

$$\begin{cases} x_i^* = 0 \\ x_i^* - \delta_i^{-1} F_i(x^*) \le 0 \end{cases} \text{ or } \begin{cases} x_i^* = x_i^* - \delta_i^{-1} F_i(x^*) \\ x_i^* - \delta_i^{-1} F_i(x^*) \ge 0 \end{cases}$$

holds. Hence, it follows from $\delta_i > 0$ that either $x_i^* = 0$ and $F_i(x^*) \ge 0$, or $x_i^* \ge 0$ and $F_i(x^*) = 0$ holds for all i. Thus x^* solves (1). □

Using the function (2), an equivalent optimization problem is obtained for nonlinear complementarity problem (1).

Proposition 1 *Let the function $f : R^n \to R$ be defined by (2). Then $f(x) \ge 0$ for all $x \ge 0$, and $f(x) = 0$ if and only if x solves (1). Hence, x solves (1) if and only if it is a solution to the following optimization problem and its optimal value is zero:*

$$\text{minimize } f(x) \text{ subject to } x \ge 0. \tag{5}$$

Proof. We first show that $f_i(x) \ge 0$ for all $x \ge 0$, so that $f(x) \ge 0$ for all $x \ge 0$. If $F_i(x) - \delta_i x_i \le 0$, then $f_i(x) = (2\delta_i)^{-1} F_i(x)^2 \ge 0$. So we consider the case $F_i(x) - \delta_i x_i > 0$. Since $\delta_i > 0$, $x_i \ge 0$ and $F_i(x) > \delta_i x_i$ hold, we see, from (3),

$$\begin{aligned} f_i(x) &= \frac{1}{2\delta_i}\{F_i(x)^2 - (F_i(x) - \delta_i x_i)^2\} \\ &= x_i F_i(x) - \frac{\delta_i}{2} x_i^2 \\ &\ge \frac{\delta_i}{2} x_i^2 \\ &\ge 0. \end{aligned}$$

Therefore, $f(x) \ge 0$ for all $x \ge 0$.

Next, suppose $f(x) = 0$. Then $f_i(x) = 0$ must hold for all i. Hence, as shown in the above, either $F_i(x) = 0$ and $F_i(x) - \delta_i x_i \le 0$, or $x_i = 0$ and $F_i(x) - \delta_i x_i > 0$ holds for all i. Therefore, x is a solution of (1).

On the other hand, suppose that x solves (1). Then either $F(x_i) = 0$ or $x_i = 0$ holds for all i. If $F(x_i) = 0$, then from (3) we have

$$f_i(x) = -\frac{1}{2\delta_i}(\max(0, -\delta_i x_i))^2 = 0.$$

Also if $x_i = 0$, then we have

$$\begin{aligned} f_i(x) &= \frac{1}{2\delta_i}\{F_i(x)^2 - (\max(0, F_i(x)))^2\} \\ &= \frac{1}{2\delta_i}\{F_i(x)^2 - F_i(x)^2\} \\ &= 0. \end{aligned}$$

Therefore, $f(x) = 0$. □

Remark 1 When problem (1) has no solution, the optimization problem (5) may have a minimizer which does not zero the function f. For example, let consider the case R^1 and $F(x) = -x - 1$. Clearly, the complementarity problem has no solution. On the other hand, given $\delta > 0$, the corresponding optimization problem (5) is formulated as

$$\text{minimize} \quad \frac{1}{2\delta}(x+1)^2 \quad \text{subject to} \quad x \geq 0.$$

The unique optimal solution to this problem is $x = 0$, at which the function value is $\frac{1}{2\delta} > 0$.

It can be shown that the function f is continuously differentiable whenever so is the mapping F.

Proposition 2 *If the mapping F is continuously differentiable, then so is the function f defined by (2), and the gradient of f is given by*

$$\nabla f(x) = F(x) - (\nabla F(x) - D)(M(x) - x). \tag{6}$$

Proof. We first note that, if a function $\theta : R^n \to R$ is continuously differentiable and $\Theta(x) = (\max(0, \theta(x)))^2$, then Θ is continuously differentiable and the gradient of Θ is given by

$$\nabla\Theta(x) = 2\max(0, \theta(x))\nabla\theta(x).$$

Hence, from (3) we have

$$\nabla f_i(x) = \frac{1}{\delta_i}(F_i(x) - \max(0, F_i(x) - \delta_i x_i))\nabla F_i(x) + \max(0, F_i(x) - \delta_i x_i)e_i. \tag{7}$$

Since

$$\begin{aligned} \max(0, F_i(x) - \delta_i x_i) &= F_i(x) - \delta_i x_i + \delta_i \max(0, x_i - \delta_i^{-1} F_i(x)) \\ &= F_i(x) - \delta_i x_i + \delta_i M_i(x) \end{aligned}$$

holds, we have from (7)

$$\begin{aligned} \nabla f_i(x) &= \frac{1}{\delta_i}(F_i(x) - \max(0, F_i(x) - \delta_i x_i))\nabla F_i(x) + \max(0, F_i(x) - \delta_i x_i)e_i \\ &= (x_i - M_i(x))\nabla F_i(x) + (F_i(x) - \delta_i x_i + \delta_i M_i(x))e_i. \end{aligned} \tag{8}$$

Therefore, we have from (8)

$$\begin{aligned} \nabla f(x) &= \sum_{i=1}^{n} \nabla f_i(x) \\ &= \sum_{i=1}^{n} \{(x_i - M_i(x))\nabla F_i(x) + (F_i(x) - \delta_i x_i + \delta_i M_i(x))e_i\} \\ &= F(x) - (\nabla F(x) - D)(M(x) - x). \end{aligned}$$

□

Proposition 1 says that finding a global optimal solution to (5) amounts to solving the complementarity problem (1). However, in general, optimization algorithms may only find a stationary point of the problem. Thus it is desirable to clarify conditions under which any stationary point of problem (5) actually solves problem (1). The next proposition answers for this question.

Proposition 3 *Suppose that $\nabla F(x)$ is positive definite for all $x \geq 0$. If $x \geq 0$ is a stationary point of problem* (5), *i.e.,*

$$(y-x)^T \nabla f(x) \geq 0 \quad \text{for all} \quad y \geq 0, \tag{9}$$

then x is a global optimal solution of problem (5), *and hence it solves the nonlinear complementarity problem* (1).

Proof. Suppose that x satisfies (9). Then from (6) we have

$$\begin{aligned}
&(M(x)-x)^T \nabla f(x) \\
&\quad = (M(x)-x)^T \{(F(x) - Dx + DM(x)) - \nabla F(x)(M(x)-x)\} \\
&\quad = (M(x)-x)^T (F(x) - Dx + DM(x)) - (M(x)-x)^T \nabla F(x)(M(x)-x).
\end{aligned} \tag{10}$$

It is easy to see that

$$\begin{aligned}
&(M(x)-x)^T(F(x) - Dx + DM(x)) \\
&\quad = \sum_{i=1}^{n} \{\delta_i(\max(0, x_i - \delta_i^{-1}F_i(x)) - x_i)^2 + (\max(0, x_i - \delta_i^{-1}F_i(x)) - x_i)F_i(x)\} \\
&\quad \leq 0.
\end{aligned} \tag{11}$$

Since x satisfies (9), it follows from (10) and (11) that

$$(M(x)-x)^T \nabla F(x)(M(x)-x) \leq 0.$$

However, since $\nabla F(x)$ is positive definite, we have $x = M(x)$. Therefore, it follows from Lemma 1 that x is a solution to (1). □

We say that the mapping F is strongly monotone on R^n_+ with modulus $\mu > 0$, if

$$(x-y)^T(F(x) - F(y)) \geq \mu \parallel x - y \parallel^2 \quad \text{for all} \quad x, y \geq 0. \tag{12}$$

The following result establishes an asymptotic behavior of the function f. Note that similar results have not been obtained for variational inequality problems.

Proposition 4 *If F is strongly monotone on R^n_+, then*

$$\lim_{x \geq 0,\ \|x\| \to \infty} f(x) = +\infty.$$

Proof. Let $\{x^k\}$ be a sequence such that $x^k \geq 0$ and $\| x^k \| \to \infty$. Taking a subsequence if necessary, we may suppose that there exists a set $I \subset \{1, 2, \ldots, n\}$ such that $x_i^k \to +\infty$ for $i \in I$ and $\{x_i^k\}$ is bounded for $i \notin I$. From $\{x^k\}$, we define another sequence $\{y^k\}$ such that $y_i^k = 0$ if $i \in I$ and $y_i^k = x_i^k$ if $i \notin I$. From (12) and the definition of y^k, we have

$$\sum_{i \in I}(F_i(x^k) - F_i(y^k))x_i^k \geq \mu \sum_{i \in I}(x_i^k)^2. \tag{13}$$

By Cauchy's inequality we have

$$\left(\sum_{i \in I}(F_i(x^k) - F_i(y^k))^2\right)^{\frac{1}{2}} \left(\sum_{i \in I}(x_i^k)^2\right)^{\frac{1}{2}} \geq \sum_{i \in I}(F_i(x^k) - F_i(y^k))x_i^k. \tag{14}$$

It then follows from (13) and (14) that

$$\sum_{i\in I}(F_i(x^k) - F_i(y^k))^2 \geq \mu^2 \sum_{i\in I}(x_i^k)^2. \tag{15}$$

Since $\{y^k\}$ is bounded by definition, $\{F_i(y^k)\}$ is also bounded. Therefore, since $x_i^k \to +\infty$ for all $i \in I$, (15) implies

$$\sum_{i\in I} F_i(x^k)^2 \to \infty.$$

As shown at the beginning of the proof of Proposition 1, $f_i(x^k) = \frac{1}{2\delta_i}F_i(x^k)^2 \geq 0$ if $F_i(x^k) - \delta_i x_i^k \leq 0$, and $f_i(x^k) \geq \frac{\delta_i}{2}(x_i^k)^2$ if $F_i(x^k) - \delta_i x_i^k > 0$. Therefore we have

$$\begin{aligned} f(x^k) &= \sum_{i=1}^{n} f_i(x^k) \\ &\geq \sum_{i\in I} f_i(x^k) \\ &\geq \sum_{i\in I} \frac{1}{2\delta_i} \min(F_i(x^k)^2, (\delta_i x_i^k)^2). \end{aligned}$$

Since $x_i^k \to +\infty$ for all $i \in I$ and $\sum_{i\in I} F_i(x^k)^2 \to \infty$, it follows that $f(x^k) \to +\infty$. □

3. Algorithms

In this section, we propose two globally convergent methods for solving (5). One is based on the methods of Fukushima [7] and the other the method of Taji et al. [17], which were originally proposed for variational inequality problems. Throughout this section, we suppose that the mapping F is strongly monotone on R_+^n with modulus $\mu > 0$.

3.1. Descent method

The first method uses the vector

$$\begin{aligned} d &= M(x) - x \\ &= \max(0, x - D^{-1}F(x)) - x \end{aligned} \tag{16}$$

as a search direction at x. When the mapping F is strongly monotone, it can be shown that the vector d given by (16) is a descent direction.

Lemma 2 *If F is strongly monotone with modulus μ, then the vector d given by* (16) *satisfies the descent condition*

$$\nabla f(x)^T d \leq -\mu \left\| d \right\|^2 .$$

Proof. From (10) and (11), we have

$$(M(x) - x)^T \nabla f(x) \leq -(M(x) - x)^T \nabla F(x)(M(x) - x). \tag{17}$$

It is known that, when F is differentiable and strongly monotone on R_+^n, ∇F satisfies

$$(y - x)^T \nabla F(x)(y - x) \geq \mu \parallel y - x \parallel^2 \text{ for all } x, y \geq 0.$$

Therefore, from (17) we have

$$\nabla f(x)^T d \leq -\mu \, \| d \|^2 . \qquad \square$$

Thus the direction d can be used to determine the next iterate by using the following Armijo-type line search rule: Let $\alpha := \beta^{\hat{m}}$, where $\hat{m}$ is the smallest nonnegative integer m such that

$$f(x) - f(x + \beta^m d) \geq \sigma \beta^m \, \| d \|^2,$$

where $0 < \beta < 1$ and $\sigma > 0$. Note that, in the descent method originally proposed by Fukushima [7] for variational inequality problems, the line search only examines step sizes shorter than unity. Here, we propose the algorithm that allows longer step sizes at each iteration.

Algorithm 1:

Choose $x^0 \geq 0$, $\alpha > 1$, $\beta \in (0, 1)$, $\sigma > 0$ and a positive diagonal matrix D;
$k := 0$
while convergence criterion is not satisfied **do**
    $d^k := \max(0, x^k - D^{-1}F(x^k)) - x^k$;
    $\hat{t} := \max\{t \mid x^k + td^k \geq 0, \, t \geq 0\}$;
    $m := 0$
    if $f(x^k) - f(x^k + d^k) \geq \sigma \, \| d^k \|^2$ **then**
        while $\alpha^m \leq \hat{t}$ and $f(x^k) - f(x^k + \alpha^m d^k) \geq \sigma \alpha^m \, \| d^k \|^2$
            and $f(x^k + \alpha^{m+1} d^k) \leq f(x^k + \alpha^m d^k)$ **do**
            $m := m + 1$
        endwhile
        $x^{k+1} := x^k + \alpha^m d^k$
    else
        while $f(x^k) - f(x^k + \beta^m d^k) < \sigma \beta^m \, \| d^k \|^2$ **do**
            $m := m + 1$
        endwhile
        $x^{k+1} := x^k + \beta^m d^k$
    endif
    $k := k + 1$
endwhile

Note that the vector $M(x^{k+1}) = \max(0, x^{k+1} - D^{-1}F(x^{k+1}))$ has already been found at the previous iteration as a by-product of evaluating f. Therefore one need not compute again the search direction d^k at the beginning of each iteration.

Theorem 1 *Suppose that the mapping F is continuously differentiable and strongly monotone with modulus μ on R^n_+. Suppose also that ∇F is Lipschitz continuous on any bounded subset of R^n_+. Then, for any starting point $x^0 \geq 0$, the sequence $\{x^k\}$ generated by Algorithm 1 converges to the unique solution of problem (1) if the positive constant σ is chosen to be sufficiently small such that $\sigma < \mu$.*

Proof. By Proposition 4, the level set $S = \{x \mid f(x) \leq f(x^0)\}$ is bounded. Hence ∇F is Lipschitz continuous on S. Since F is continuously differentiable, it is easy to show that F

is also Lipschitz continuous on S. Under these conditions, it is not difficult to show that ∇f is Lipschitz continuous on S, i.e., there exists a constant $L > 0$ such that

$$\| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \| \quad \text{for all} \quad x, y \in S.$$

Therefore, as shown in the proof of [7, Theorem 4.2], any accumulation point of $\{x^k\}$ satisfies $x = M(x)$, and hence solves (1) by Lemma 1. Since strong monotonicity of F ensures that problem (1) has a unique solution, we can conclude that the entire sequence converges to the unique solution of (1). □

Remark 2 In Fukushima [7], global convergence theorem assumes not only the strong monotonicity of mapping F but the *compactness* of the constraint set, which is not the case for nonlinear complementarity problems. Theorem 3.1 above establishes global convergence under strong monotonicity of the mapping F only.

3.2. Modification of Newton method

The second method is a modification of Newton method, which incorporates a line search strategy. The original Newton method for solving the nonlinear complementarity problem (1) generates a sequence $\{x^k\}$ such that $x^0 \geq 0$ and x^{k+1} is determined as $x^{k+1} := \bar{x}$, where $\bar{x}$ is a solution to the following linearized complementarity problem:

$$x \geq 0, F(x^k) + \nabla F(x^k)^T(x - x^k) \geq 0 \text{ and } x^T(F(x^k) + \nabla F(x^k)^T(x - x^k)) = 0. \tag{18}$$

It is shown [16] that, under suitable assumptions, the sequence generated by (18) quadratically converges to a solution x^* of the problem (1), provided that the starting point x^0 is chosen sufficiently close to x^*.

Lemma 3 *When the mapping F is strongly monotone with modulus μ, the vector $d^k := \bar{x} - x^k$ obtained by solving the linearized complementarity problem* (18) *satisfies the inequality*

$$\nabla f(x^k)^T d^k < -\left(\mu - \frac{1}{4} \| D \|\right) \| d^k \|^2 .$$

Therefore, d^k is actually a feasible descent direction of f at x^k, if the matrix D is chosen to satisfy $\| D \| = \max_i(\delta_i) < 4\mu$.

Proof. For simplicity of presentation, we omit the superscript k in x^k and d^k. Since $d := \bar{x} - x$, it follows from (6) that

$$\begin{aligned} \nabla f(x)^T d &= (\bar{x} - x)^T F(x) + (\bar{x} - x)^T (\nabla F(x) - D)(x - M(x)) \\ &= (\bar{x} - x)^T (F(x) + \nabla F(x)^T (\bar{x} - x)) - (\bar{x} - x)^T \nabla F(x)^T (\bar{x} - x)) \\ &\quad + (x - M(x))^T (F(x) + \nabla F(x)^T (\bar{x} - x)) \\ &\quad - (x - M(x))^T F(x) - (\bar{x} - x)^T D(x - M(x)) \\ &= -(M(x) - \bar{x})^T (F(x) + \nabla F(x)^T (\bar{x} - x)) \\ &\quad + (M(x) - x)^T F(x) + (\bar{x} - x)^T D(M(x) - x) \\ &\quad - (\bar{x} - x)^T \nabla F(x)^T (\bar{x} - x). \end{aligned} \tag{19}$$

Since $\bar{x}$ is a solution to (18) and $M(x) \geq 0$, the first term of (19) is nonpositive. From (11), we have

$$(M(x) - x)^T F(x) \leq -(M(x) - x)^T D(M(x) - x).$$

Then it follows from the second term of (19) that

$$\begin{aligned}
&(M(x)-x)^T F(x) + (\bar{x}-x)^T D(M(x)-x) \\
&\le (\bar{x}-x)^T D(M(x)-x) - (M(x)-x)^T D(M(x)-x) \\
&= \sum_{i=1}^{n} \delta_i \{(\bar{x}_i - x_i)(M_i(x)-x_i) - (M_i(x)-x_i)^2\} \\
&= \sum_{i=1}^{n} \frac{\delta_i}{2}\{(\bar{x}_i - x_i)^2 - (M_i(x)-x_i)^2 - (\bar{x}_i - M_i(x))^2\} \\
&\le \sum_{i=1}^{n} \frac{\delta_i}{2}\left\{(\bar{x}_i - x_i)^2 - \frac{1}{2}(\bar{x}_i - x_i)^2\right\} \\
&= \sum_{i=1}^{n} \frac{\delta_i}{4}(\bar{x}_i - x_i)^2 \\
&= \frac{1}{4}(\bar{x}-x)^T D(\bar{x}-x). \qquad (20)
\end{aligned}$$

Hence, we have from (19) and (20)

$$\nabla f(x)^T d \le -d^T \nabla F(x)^T d + \frac{1}{4} d^T D d.$$

But since strong monotonicity of F implies $d^T \nabla F(x) d \ge \mu \, \| d \|^2$ and since $d^T D d \le \| D \| \, \| d \|^2$, we have

$$\nabla f(x)^T d < -\left(\mu - \frac{1}{4} \| D \|\right) \| d \|^2 .$$

The last half of the proposition then follows immediately. □

Using this result, we can construct a modified Newton method for solving the nonlinear complementarity problem (1).

Algorithm 2:

Choose $x^0 \ge 0$, $\beta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$, and a positive diagonal matrix D;
$k := 0$
while convergence criterion is not satisfied **do**
  find $\bar{x}^k$ such that
    $\bar{x}^k \ge 0, F(x^k) + \nabla F(x^k)^T(\bar{x}^k - x^k) \ge 0$ and $(\bar{x}^k)^T(F(x^k) + \nabla F(x^k)^T(\bar{x}^k - x^k)) = 0$;
  $d^k := \bar{x}^k - x^k$;
  $m := 0$
  while $f(x^k) - f(x^k + \beta^m d^k) < -\sigma \beta^m \nabla f(x^k)^T d^k$ **do**
    $m := m + 1$
  endwhile
  $x^{k+1} := x^k + \beta^m d^k$;
  $k := k + 1$
endwhile

When the mapping F is strongly monotone, we can establish the global convergence of Algorithm 2.

Theorem 2 *Suppose that the mapping F is continuously differentiable and strongly monotone with modulus μ. If the matrix D is chosen such that $\| D \| = \max_i(\delta_i) < 4\mu$, then,* *for any starting point $x^0 \geq 0$, the sequence $\{x^k\}$ generated by Algorithm 2 converges to the solution of* (1).

Proof. By Lemma 3 and the Armijo line search rule, the sequence $\{f(x^k)\}$ is nonincreasing. It then follows from Proposition 4 that the sequence $\{x^k\}$ is bounded, and hence it contains at least one accumulation point. As shown in the proof of [17, Theorem 4.1], any accumulation point of $\{x^k\}$ is a solution of (1). Since strong monotonicity of F ensures that problem (1) has a unique solution, we can conclude that the entire sequence converges to the unique solution of (1). □

We can also show that the rate of convergence of Algorithm 2 is quadratic if $F \in C^2$ and the strict complementarity condition holds at the unique solution x^* of (1).

Theorem 3 *Suppose that the sequence $\{x^k\}$ generated by Algorithm 2 converges to the solution x^* to problem* (1). *Suppose also that the mapping F belongs to class C^2, $\nabla F(x^*)$ is positive definite and $\nabla^2 F$ is Lipschitz continuous on some neighborhood of x^*. If the strict complementarity condition holds at x^*, i.e., $x_i^* = 0$ implies $F_i(x^*) > 0$ for all $i = 1, 2, \ldots, n$, then there exists an integer $\bar{k}$ such that the unit step size is accepted for all $k \geq \bar{k}$. Therefore, the sequence $\{x^k\}$ converges quadratically to the solution x^*.*

Before proving Theorem 3, we show the following lemma.

Lemma 4 *Let x^* be a solution to problem* (1). *If $F \in C^2$ and the strict complementarity condition holds at x^*, then $f_i \in C^2$ on a neighborhood of x^*, and the gradient and the Hessian of f_i are given by*

$$\nabla f_i(x) = \begin{cases} (x_i \nabla F_i(x) + F_i(x) e_i) - \delta_i x_i e_i & \text{if } i \in I^* \quad (21a) \\ \frac{1}{\delta_i} F_i(x) \nabla F_i(x) & \text{if } i \in \bar{I}^* \quad (21b) \end{cases}$$

and

$$\nabla^2 f_i(x) = \begin{cases} x_i \nabla^2 F_i(x) + 2\nabla F_i(x) e_i^T - \delta_i e_i e_i^T & \text{if } i \in I^* \quad (22a) \\ \frac{1}{\delta_i}(F_i(x) \nabla^2 F_i(x) + \nabla F_i(x) \nabla F_i(x)^T) & \text{if } i \in \bar{I}^*, \quad (22b) \end{cases}$$

respectively, where $I^* = \{i \mid x_i^* = 0\}$ and $\bar{I}^* = \{i \mid x_i^* > 0\}$.

Proof. The strict complementarity and the continuity of F ensure that there is a neighborhood B of x^* such that

$$\begin{cases} F_i(x) - \delta_i x_i > 0 & \text{if } i \in I^* \\ F_i(x) - \delta_i x_i < 0 & \text{if } i \in \bar{I}^* \end{cases} \tag{23}$$

holds for all $x \in B$. Hence, we have from (3)

$$f_i(x) = \begin{cases} x_i F_i(x) - \frac{\delta_i}{2} x_i^2 & \text{if } i \in I^* \quad (24a) \\ \frac{1}{2\delta_i} F_i(x)^2 & \text{if } i \in \bar{I}^*. \quad (24b) \end{cases}$$

Therefore, by differentiating (24a) and (24b) directly, we have (21a), (21b) and (22a), (22b). □

Proof of Theorem 3. It is sufficient to show that $f(x^k) - f(\bar{x}^k) \geq -\sigma \nabla f(x^k)^T (\bar{x}^k - x^k)$ holds for sufficiently large k. For simplicity, we consider the case $\delta_1 = \cdots = \delta_n = \delta > 0$, i.e., the diagonal matrix D is the identity matrix multiplied by $\delta > 0$. It is not difficult to extend the result to the general case. Without loss of generality, we assume $\bar{I}^* = \{l, l+1, \ldots, n\}$, where $1 \le l \le n$, and denote

$$x = \begin{pmatrix} x_{I^*} \\ x_{\bar{I}^*} \end{pmatrix}, \quad F(x) = \begin{pmatrix} F_{I^*}(x) \\ F_{\bar{I}^*}(x) \end{pmatrix}, \quad \nabla F(x) = \begin{pmatrix} \nabla_{I^*} F_{I^*}(x) & \nabla_{I^*} F_{\bar{I}^*}(x) \\ \nabla_{\bar{I}^*} F_{I^*}(x) & \nabla_{\bar{I}^*} F_{\bar{I}^*}(x) \end{pmatrix}.$$

Since the strict complementarity holds at x^*, there is an integer K_1 such that x^k satisfies

$$\begin{cases} F_i(x^k) - \delta x_i^k > 0 & \text{if } i \in I^* \quad (25a) \\ F_i(x^k) - \delta x_i^k < 0 & \text{if } i \in \bar{I}^* \quad (25b) \end{cases}$$

for all $k \ge K_1$. Under the given assumptions, Newton method (18) is locally quadratically convergent to the solution x^* [16]. Hence, it follows from the strict complementarity and the continuity of F and ∇F that there is an integer K_2 such that

$$\begin{cases} \bar{x}_i^k = 0 & \text{and} \quad F_i(x^k) + \nabla F_i(x^k)^T(\bar{x}^k - x^k) > 0 \quad \text{if} \quad i \in I^* \quad (26a) \\ \bar{x}_i^k > 0 & \text{and} \quad F_i(x^k) + \nabla F_i(x^k)^T(\bar{x}^k - x^k) = 0 \quad \text{if} \quad i \in \bar{I}^* \quad (26b) \end{cases}$$

for all $k \ge K_2$.

Now suppose $k \ge \max(K_1, K_2)$. For simplicity of presentation, we omit superscript k in x^k and $\bar{x}^k$. For each $i \in I^*$, we have

$$\begin{aligned} & f_i(x) - f_i(\bar{x}) + \sigma \nabla f_i(x)^T(\bar{x} - x) \\ = & \left(x_i F_i(x) - \frac{\delta}{2} x_i^2\right) - \left(\bar{x}_i F_i(\bar{x}) - \frac{\delta}{2}\bar{x}_i^2\right) + \sigma(x_i \nabla F_i(x) + F_i(x) e_i - \delta x_i e_i)^T(\bar{x} - x) \\ = & \ x_i F_i(x) - \frac{\delta}{2} x_i^2 + \sigma(x_i \nabla F_i(x)^T(\bar{x} - x) - x_i F_i(x) + \delta x_i^2) \\ \ge & \ x_i F_i(x) - \frac{\delta}{2} x_i^2 + \sigma(-2 x_i F_i(x) + \delta x_i^2) \\ = & \ (1 - 2\sigma)\left(x_i F_i(x) - \frac{\delta}{2} x_i^2\right) \\ \ge & \left(\frac{1}{2} - \sigma\right) \delta x_i^2 \\ = & \left(\frac{1}{2} - \sigma\right) \delta (\bar{x}_i - x_i)^2, \end{aligned} \tag{27}$$

where the first equality follows from (21a) and (24a), the second equality and the first inequality follow from (26a), the second inequality follows from (25a) and the last equality follows from (26a).

On the other hand, since by the mean value theorem we have

$$f_i(\bar{x}) - f_i(x) = \nabla f_i(x)^T(\bar{x} - x) + \frac{1}{2}(\bar{x} - x)^T \nabla^2 f_i(\xi)(\bar{x} - x)$$

for some ξ in line segment of x and $\bar{x}$, we have

$$\begin{aligned} & f_i(x) - f_i(\bar{x}) + \sigma \nabla f_i(x)^T(\bar{x} - x) \\ & = (\sigma - 1)\nabla f_i(x)^T(\bar{x} - x) - \frac{1}{2}(\bar{x} - x)^T \nabla^2 f_i(\xi)(\bar{x} - x) \\ & = (\sigma - 1)\nabla f_i(x)^T(\bar{x} - x) - \frac{1}{2}(\bar{x} - x)^T \nabla^2 f_i(x)(\bar{x} - x) \\ & \quad + \frac{1}{2}(\bar{x} - x)^T(\nabla^2 f_i(x) - \nabla^2 f_i(\xi))(\bar{x} - x). \end{aligned} \tag{28}$$

Then for each $i \in \bar{I}^*$, we have

$$
\begin{aligned}
&(\sigma - 1)\nabla f_i(x)^T(\bar{x} - x) - \frac{1}{2}(\bar{x} - x)^T\nabla^2 f_i(x)(\bar{x} - x)\\
&= \frac{\sigma - 1}{\delta}F_i(x)\nabla F_i(x)^T(\bar{x} - x) - \frac{1}{2\delta}(\bar{x} - x)^T(F_i(x)\nabla^2 F_i(x) + \nabla F_i(x)\nabla F_i(x)^T)(\bar{x} - x)\\
&= \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\nabla F_i(x)^T(\bar{x} - x))^2 - \frac{1}{2\delta}F_i(x)(\bar{x} - x)^T\nabla^2 F_i(x)(\bar{x} - x)\\
&= \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\nabla F_i(x^*)^T(\bar{x} - x))^2 - \frac{1}{2\delta}F_i(x)(\bar{x} - x)^T\nabla^2 F_i(x)(\bar{x} - x)\\
&\quad + \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\bar{x} - x)^T(\nabla F_i(x)\nabla F_i(x)^T - \nabla F_i(x^*)\nabla F_i(x^*)^T)(\bar{x} - x),
\end{aligned}
\tag{29}
$$

where the first equality follows from (21b) and (22b), and the second equality follows from (26b). Hence, we have from (28) and (29)

$$
\begin{aligned}
&f_i(x) - f_i(\bar{x}) + \sigma\nabla f_i(x)^T(\bar{x} - x)\\
&= \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\nabla F_i(x)^T(\bar{x} - x))^2 - \frac{1}{2\delta}F_i(x)(\bar{x} - x)^T\nabla^2 F_i(x)(\bar{x} - x)\\
&\quad + \frac{1}{2}(\bar{x} - x)^T(\nabla^2 f_i(x) - \nabla^2 f_i(\xi))(\bar{x} - x)\\
&\quad + \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\bar{x} - x)^T(\nabla F_i(x)\nabla F_i(x)^T - \nabla F_i(x^*)\nabla F_i(x^*)^T)(\bar{x} - x).
\end{aligned}
$$

When $\nabla^2 F$ is Lipschitz continuous and is bounded on some neighborhood of x^*, it is not difficult to show that $\nabla^2 f_i$ and ∇F_i are also Lipschitz continuous. Moreover, for $i \in \bar{I}^*$ we have $F_i(x) \to 0$ if $x \to x^*$. Hence, we have

$$
\begin{aligned}
&f_i(x) - f_i(\bar{x}) + \sigma\nabla f_i(x)^T(\bar{x} - x)\\
&\geq \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\nabla F_i(x^*)^T(\bar{x} - x))^2 + O(\| x - x^* \| + \| \bar{x} - x \|) \| \bar{x} - x \|^2
\end{aligned}
\tag{30}
$$

holds on some neighborhood of x^*.

Therefore, it follows from (27) and (30) that

$$
\begin{aligned}
&f(x) - f(\bar{x}) + \sigma\nabla f(x)^T(\bar{x} - x)\\
&\geq \delta\left(\frac{1}{2} - \sigma\right)\sum_{i\in I^*}(\bar{x}_i - x_i)^2 + \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)\sum_{i\in \bar{I}^*}(\nabla F_i(x)^T(\bar{x} - x))^2\\
&\quad + O(\| x - x^* \| + \| \bar{x} - x \|) \| \bar{x} - x \|^2\\
&= \frac{1}{\delta}\left(\frac{1}{2} - \sigma\right)(\bar{x} - x)^T A(\bar{x} - x) + O(\| x - x^* \| + \| \bar{x} - x \|) \| \bar{x} - x \|^2,
\end{aligned}
\tag{31}
$$

where A is a matrix such that

$$
\begin{aligned}
A &= \begin{pmatrix} \delta^2 E_l & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & \nabla_{I^*}F_{\bar{I}^*}(x^*) \\ 0 & \nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*) \end{pmatrix} \begin{pmatrix} 0 & 0 \\ \nabla_{I^*}F_{\bar{I}^*}(x^*)^T & \nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*)^T \end{pmatrix}\\
&= \begin{pmatrix} \delta E_l & \nabla_{I^*}F_{\bar{I}^*}(x^*) \\ 0 & \nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*) \end{pmatrix} \begin{pmatrix} \delta E_l & \nabla_{I^*}F_{\bar{I}^*}(x^*) \\ 0 & \nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*) \end{pmatrix}^T
\end{aligned}
$$

where E_l is the $l \times l$ identity matrix. Clearly A is positive semi-definite. Moreover, since $\nabla F(x^*)$ is positive definite by assumption, the matrix $\nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*)$ is also positive definite. Hence, the matrix $\begin{pmatrix} \delta E_l & \nabla_{I^*}F_{\bar{I}^*}(x^*) \\ 0 & \nabla_{\bar{I}^*}F_{\bar{I}^*}(x^*) \end{pmatrix}$ is nonsingular , and hence A is positive definite.

Therefore, (31) is strictly positive provided that x is sufficiently close to x^*. □

Remark 3 Taji et al. [17] have obtained a globally convergent Newton method for variational inequality problems. In their method, to obtain quadratic convergence, the following line search procedure is used:

Let $0 < \beta < 1$, $0 < \gamma < 1$ and $\sigma \in (0,1)$;
if $f(x^k + d^k) \le \gamma f(x^k)$ **then**
$\alpha_k := 1$
else
$m := 0$
while $f(x^k) - f(x^k + \beta^m d^k) < -\sigma\beta^m {d^k}^T \nabla f(x^k)$ **do**
$m := m + 1$
endwhile
$\alpha_k := \beta^m$
endif
$x^{k+1} := x^k + \alpha_k d^k$

Note that this line search procedure, which is similar to the one used by Marcotte and Dussault [14], first checks if the unit step size is acceptable. On the other hand, Algorithm 2 employs Armijo rule in a more direct manner.

4. Computational results I

In the following two sections, we report some numerical results for Algorithms 1 and 2 discussed in the previous section. In this section, we present the results for a strongly monotone problem. All computer programs were coded in FORTRAN and the runs in this section were made in double precision on a personal computer called Fujitsu FMR-70.

Throughout the computational experiments, the parameters used in the algorithms were set as $\alpha = 2$, $\beta = 0.5$, $\gamma = 0.5$ and $\sigma = 0.0001$. The positive diagonal matrix D was chosen to be the identity matrix multiplied by a positive parameter $\delta > 0$. Therefore the merit function (2) can be written simply as

$$f(x) = \frac{1}{2\delta}\sum_{i=1}^{n}\left\{F_i(x)^2 - (\max(0, F_i(x) - \delta x_i))^2\right\}. \tag{32}$$

The search direction of Algorithms 1 can also be written as

$$d^k := \max\left(0, x^k - \frac{1}{\delta}F(x^k)\right) - x^k.$$

The convergence criterion was

$$|\min(x_i, F_i(x))| \le 10^{-5} \text{ for all } i = 1, 2, \ldots, n.$$

For comparison purposes, we also tested two popular methods for solving the nonlinear complementarity problem, the projection method [4] and Newton method without line search [12]. The projection method generates a sequence $\{x^k\}$ such that $x^0 \ge 0$ and x^{k+1} is determined from x^k by

$$x^{k+1} := \max\left(0, x^k - \frac{1}{\delta}F(x^k)\right), \tag{33}$$

for all k. Note that this method may be considered a fixed step-size variant of Algorithm 1. When the mapping F is strongly monotone and Lipschitz continuous with constants μ

and L, respectively, this method is globally convergent if δ is chosen large enough to satisfy $\delta > L^2/2\mu$ (see [16, Corollary 2.11.]).

The mappings used in this section are of the form

$$F(x) = Ix + \rho(N - N^T)x + \phi(x) + c, \tag{34}$$

where I is the $n\times n$ identity matrix, N is an $n\times n$ matrix such that each row contains only one nonzero element, and $\phi(x)$ is a nonlinear monotone mapping with components $\phi_i(x_i) = p_i x_i^4$, where p_i are positive constants. Elements of matrix N and vector c as well as coefficients p_i are randomly generated such that $-5 \le N_{ij} \le 5$, $-25 \le c_i \le 25$ and $0.001 \le p_i \le 0.006$. The results are shown in Tables 1 $\sim$ 4. All starting points were chosen to be $(0, 0, \ldots, 0)$. In the tables, $\#f$ is the total number of evaluating the merit function f and all CPU times are in seconds and exclude input/output times. The parameter ρ is used to change the degree of asymmetry of F, namely F deviates from symmetry as ρ becomes large. Since the matrix $I + \rho(N - N^T)$ is positive definite for any ρ and $\phi_i(x_i)$ are monotonically increasing for $x_i \ge 0$, the mapping F defined by (34) is strongly monotone on R^n_+.

Table 1: Comparison of Algorithm 1 and the projection method ($n = 10$, $\rho = 1$).

	Algorithm 1			projection method*	
δ	#Iterations	$\#f$	CPU	#Iterations	CPU
0.1	1380	9683	18.45	–	–
0.5	307	1527	3.04	–	–
1	328	1305	2.63	–	–
2	338	1009	2.13	–	–
3	353	951	2.04	–	–
4	342	685	1.55	–	–
5	256	511	1.16	–	–
6	351	701	1.72	–	–
6.2	337	674	1.66	–	–
6.3	377	754	1.91	9118	13.94
6.5	376	752	1.93	1594	2.43
7	385	770	2.07	610	0.93
8	337	676	1.98	338	0.51
9	270	542	1.59	271	0.43
10	242	487	1.41	244	0.36
12	232	468	1.36	229	0.34
15	239	488	1.41	239	0.35
20	254	636	1.72	272	0.41
50	229	920	2.17	549	0.79
100	229	1149	2.63	1036	1.47
200	239	1385	3.05	2008	2.88
500	239	1674	3.59	5007	7.08
1000	372	2605	5.55	9998	14.18

*The projection method failed to converge for the value of δ up to 6.2.

4.1. Comparison of Algorithm 1 and the projection method

First we have compared Algorithms 1 and the projection method (33) by using a 10-dimensional example, in which mapping F is given by

$$F(x) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -2 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 1 & 0 & -5 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -5 & 0 & 0 & 0 & 0 & 5 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 & 4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix} + \begin{pmatrix} 0.004x_1^4 \\ 0.004x_2^4 \\ 0.003x_3^4 \\ 0.003x_4^4 \\ 0.006x_5^4 \\ 0.006x_6^4 \\ 0.004x_7^4 \\ 0.004x_8^4 \\ 0.004x_9^4 \\ 0.002x_{10}^4 \end{pmatrix} + \begin{pmatrix} 2 \\ 10 \\ 2 \\ 9 \\ -15 \\ 12 \\ -9 \\ 5 \\ 7 \\ -17 \end{pmatrix}.$$

The results for this problem are shown in Table 1.

In general, the projection method is guaranteed to converge, only if the parameter δ is chosen sufficiently large. In fact, Table 1 shows that when δ is large, the projection method is always convergent, but as δ becomes small, the behavior of the method found to be unstable and eventually it fails to converge.

Table 1 also shows that Algorithm 1 is always convergent even if δ is chosen small, since the line search determines an adequate step size at each iteration. In Algorithm 1, the number of iterations is almost constant. This is because we may choose a larger step size when the magnitude of vector d^k is small, i.e. δ is large.

Algorithm 1 spends more CPU times per iteration than the projection method, because the former algorithm requires overheads of evaluating the merit function f. But, when δ becomes large, Algorithm 1 tends to spend less CPU time than the projection method, because the number of iterations of Algorithm 1 increases mildly.

4.2. Comparison of Algorithm 2 and Newton method

Next we have compared Algorithm 2 and the pure Newton method (18) without line search. For each of the problem sizes $n = 30, 50$ and 90, we randomly generated five test problems. The parameters ρ and δ were set as $\rho = 1$ and $\delta = 1$. The starting point was chosen to be $x = 0$. In solving the linearized subproblem at each iteration of Algorithm 2 and Newton method, we used Lemke's complementarity pivoting method [13] coded by Fukushima [11]. All parameters and starting points are set to the default values used in [11]. The results are given in Table 2. All numbers shown in Table 2 are averages of the results for five problems for each case and #Lemke is the total number of pivotings in Lemke's method.

Table 2: Comparison of Algorithm 2 and Newton method ($\rho = 1$).

	Algorithm 2				Newton method		
n	#Iterations	$\#f$	#Lemke	CPU	#Iterations	#Lemke	CPU
30	5.6	7.6	80.6	4.294	8	115.2	5.840
50	5.6	7.6	156.0	19.880	8	216.4	26.142
90	6.0	8.0	275.2	105.400	8	358.8	135.690

Table 2 shows that the number of iterations of Newton method is consistently larger than that of Algorithm 2 as far as the test problems used in the experiments are concerned. Therefore, since it is time consuming to solve a linear subproblem at each iteration, Algorithm 2 required less CPU time than the pure Newton method in spite of the overheads in line search. Finally we note that, the pure Newton method (18) is not guaranteed to be globally convergent, although it actually converged for all test problems reported in Table 2.

4.3. Comparison of Algorithms 1 and 2

Finally we have compared Algorithms 1 and 2. For each of the problem sizes $n = 30, 50$ and 90, we randomly generated five test problems. To see how these algorithms behave for various degrees of asymmetry of the mapping F, we have tested several values of ρ between 0.1 and 2.0. The starting point was chosen to be $x = 0$. The results are given in Table 3. All numbers shown in Table 3 are averages of the results for five test problems for each case.

Table 3: Comparison of Algorithms 1 and 2.

		Algorithm 1			Algorithm 2				
ρ	n	#Iterations	$\#f$	CPU	#Iterations	$\#f$	#Lemke	CPU	LEMKE
0.1	30	31.6	96.0	0.672	6.0	8.4	87.4	4.956	4.094
	50	28.4	84.6	0.966	6.0	9.2	148.0	19.050	17.989
	90	38.2	114.0	2.322	6.2	8.2	259.0	99.156	96.721
0.2	30	40.0	119.6	0.838	6.2	8.2	92.6	4.872	4.340
	50	37.6	110.0	1.264	6.4	8.8	162.6	20.888	19.633
	90	40.4	120.4	2.448	6.6	8.6	277.2	106.672	103.201
0.3	30	33.8	99.2	0.690	6.4	8.4	90.8	4.812	4.258
	50	39.2	112.2	1.292	6.2	8.6	157.8	20.270	19.055
	90	41.4	119.6	2.408	6.6	8.6	275.2	105.890	102.253
0.5	30	45.8	127.0	0.892	6.0	8.2	88.2	4.662	4.128
	50	58.6	161.8	1.848	6.0	8.0	157.8	20.242	19.134
	90	110.8	273.8	5.780	6.6	8.6	281.4	108.260	105.073
0.8	30	152.8	322.0	2.436	5.8	7.8	84.4	4.474	3.956
	50	290.4	586.4	7.450	6.0	8.0	160.4	20.476	19.368
	90	780.2	1557.4	33.960	6.0	8.0	269.4	103.266	100.296
1.0	30	394.2	792.4	5.270	5.6	7.6	80.6	4.294	3.784
	50	519.6	1077.6	11.068	5.6	7.6	156.0	19.880	18.833
	90	866.0	2129.6	35.880	6.0	8.0	275.2	105.400	102.107
1.5	30	1197.0	3793.2	21.518	5.4	7.4	80.0	4.266	3.752
	50	1604.0	4927.4	45.280	5.2	7.2	145.8	18.648	17.607
	90	2928.0	9777.8	158.162	6.0	8.0	275.6	105.492	102.606
2.0	30	3195.2	12694.8	69.636	5.2	7.2	79.0	4.168	3.712
	50	3842.6	15929.0	141.944	5.2	7.2	145.2	18.494	17.336
	90	4957.6	20905.0	332.510	5.8	7.8	279.0	106.514	103.618

Table 3 shows that when the mapping F is close to symmetry, Algorithm 1 converges very fast, and when the mapping becomes asymmetric, the number of iterations and CPU time of Algorithm 1 increase rapidly. On the other hand, in Algorithm 2, while the total number of pivotings of Lemke's method increases in proportion to problem size n, the number of iterations stays constant even when the problem size and the degree of asymmetry of F are varied. Hence, when the degree of asymmetry of F is relatively small, that is, when ρ is smaller than 1.0 in our test problems, Algorithm 1 requires less CPU time than Algorithm 2.

Note that, since the mapping F used in our computational experience is sparse, complexity of each iteration in Algorithm 1 is small. On the other hand, the code [11] of Lemke's method used in Algorithm 2 to solve a linear subproblem does not make use of sparsity.

Moreover, since Lemke's method is restrictive in the choice of initial points, at each iteration we must restart from a priori fixed initial point even when the iterate becomes close to a solution. Therefore, it may require a significant amount of CPU time at each iteration for large problems. (In Table 3, LEMKE is the total CPU time to solve subproblems by Lemke's method.) If a method that can make use of sparsity and can start from arbitrary point is available to solve a linear subproblem, CPU time of Algorithm 2 may decrease. (For the latter respect, the path-following method of van der Laan and Talman [19] might be useful.) The projected Gauss-Seidel method [3, pp. 397] for solving linear complementarity problem is one of such methods. In Table 4, results of Algorithm 2 using the projected Gauss-Seidel method in place of Lemke's method are given. Table 4 shows that, if the mapping F is almost symmetric, Algorithm 2 converges very fast. But Algorithm 2 fails to converge when the degree of asymmetry increased, because the projected Gauss-Seidel method failed to solve linear subproblems.

Table 4: Results for Algorithm 2 (Gauss-Seidel version).

		Algorithm 2			
ρ	n	#Iterations	$\#f$	#Gauss	CPU
0.1	30	6.0	8.4	39.2	0.274
	50	6.0	9.2	39.6	0.466
	90	6.2	8.2	48.0	0.916
0.2	30	3 of 5 failed			
	50	3 of 5 failed			
	90	failed			
$0.3 \sim 2.0$	30	failed			
	50				
	90				

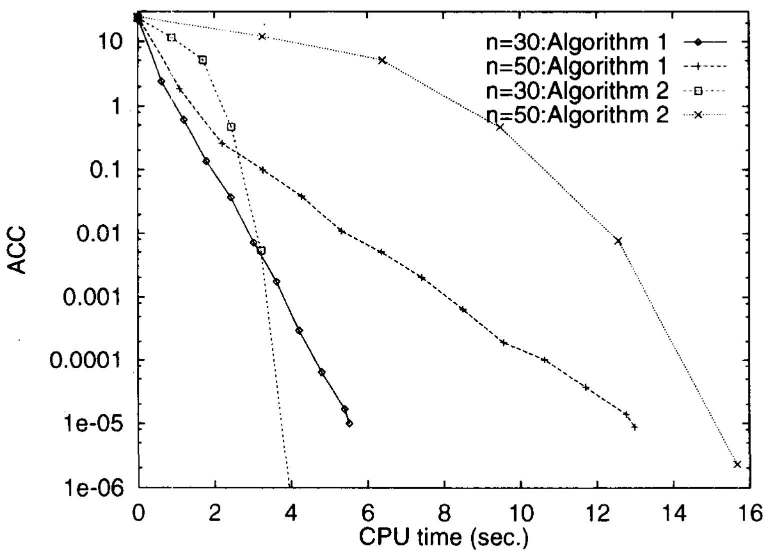

Figure 1: Behavior of Algorithms 1 and 2.

Figure 1 illustrates how Algorithms 1 and 2 converged for two typical test problems with $n = 30$ and 50. In the figure, the vertical axis represents the accuracy of a generated point to the solution, which is evaluated by

$$\text{ACC} = \max_i \{|\min(x_i, F_i(x))| \mid i = 1, 2, \ldots, n\}.$$

Figure 1 indicates that Algorithm 2 is quadratically convergent near the solution. Figure 1 also indicates that Algorithm 1 is linearly convergent though it has not been proved theoretically.

5. Computational results II

In this section, we present the results of applying Algorithms 1 and 2 to some examples which arise from an optimization problem, a spatial price equilibrium problem, a noncooperative game and a traffic assignment problem. The algorithms were implemented in FORTRAN on a SUN-4 workstation. The parameters in the algorithms were set in the same manner as in Section 4. The positive diagonal matrix D was also chosen to be the identity matrix multiplied by $\delta > 0$, and hence the merit function (32) was used. The convergence criterion was

$$|\min(x_i, F_i(x))| \leq \text{CC} \quad \text{for all } i = 1, 2, \ldots, n,$$

where CC is a parameter used to change accuracy of algorithms. In solving the linearized subproblem of Algorithm 2, we used Lemke's complementarity pivoting algorithm coded by Fukushima [11]. The results are shown in Tables 5 $\sim$ 11.

Some mappings used in the experiments were only monotone but not strongly monotone. Others were not even monotone, though they could be considered almost monotone. Thus all the problems do not satisfy the convergence conditions of our algorithms. However, for the most case both Algorithms 1 and 2 converged and produced satisfactory solutions.

Example 1 This is the following 4-variable complementarity problem from Josephy [12], whose mapping is given by

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 3x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 3x_4 - 1 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

The results are shown in Table 5. Since the mapping is co-positive but not monotone, Algorithm 2 failed when the initial points $(0, \ldots, 0)$ and $(10, \ldots, 10)$ were chosen, because the linearized subproblem at $(0, \ldots, 0)$ has no solution and the search direction at $(10, \ldots, 10)$ is not a descent direction, respectively. On the other hand, Algorithm 1 converged for those initial points.

Table 5: Results of Example 1.

		Algorithm 1			Algorithm 2			
CC	Initial point	#Iterations	#f	CPU	#Iterations	#f	#Lemke	CPU
10^{-5}	$(0, \ldots, 0)$	20	62	0.00	failed			
	$(1, \ldots, 1)$	21	63	0.00	4	5	8	0.00
	$(5, \ldots, 5)$	21	63	0.00	5	6	10	0.00
	$(10, \ldots .10)$	21	63	0.00	failed			

Example 2 This is a 10-variable complementarity problem arising from the Nash-Cournot production problem appeared in Harker [9]. In this example, the Jacobian $\nabla F(x)$ of the mapping is a P-matrix for any $x > 0$, but the mapping F is not monotone. Table 6 shows that both Algorithms 1 and 2 converged to the solution quickly.

Table 6: Results of Example 2.

		Algorithm 1			Algorithm 2			
CC	Initial point	#Iterations	#f	CPU	#Iterations	#f	#Lemke	CPU
10^{-5}	$(1,\ldots,1)$	44	88	0.08	8	10	89	0.08
	$(10,\ldots.10)$	43	86	0.08	6	7	60	0.05

Example 3 This example is the following convex programming problem:

$$\begin{array}{lll} \text{minimize} & (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^4 & \\ & +7x_6^2+2x_7^2-4x_6x_7-10x_6-8x_7 & \\ \text{subject to} & 2x_1^2+3x_2^4+x_3+4x_4^2+5x_5^2 & \leq 100 \\ & 7x_1+3x_2+10x_3^2+x_4-x_5 & \leq 200 \\ & 20x_1+x_2^2+6x_6^2-8x_7 & \leq 150 \\ & 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7 & \leq 0 \\ & x_i \geq 0,\ i=1,2,\ldots,7, & \end{array}$$

which is formulated as an 11-variable complementarity problem. Since the objective function is convex, the mapping is monotone, but not strongly monotone on R^n_+. Table 7 shows that Algorithms 1 and 2 converged for both initial points $(0,\ldots,0)$ and $(10,\ldots,10)$.

Table 7: Results of Example 3.

		Algorithm 1			Algorithm 2			
CC	Initial point	#Iterations	#f	CPU	#Iterations	#f	#Lemke	CPU
10^{-3}	$(0,\ldots,0)$	263	840	0.13	5	10	30	0.03
	$(10,\ldots.10)$	1213	4504	0.64	9	10	75	0.08
10^{-5}	$(0,\ldots,0)$	375	1119	0.19	6	11	36	0.04
	$(10,\ldots.10)$	1308	4752	0.67	10	11	81	0.08

Example 4 This example is a 15-variable traffic assignment problem from Bertsekas and Gafni [2]. This problem consists of a traffic network with 25 nodes, 40 arcs, 5 O-D pairs and 10 paths. The mapping is monotone but not strongly monotone. The results are shown in Table 8. In this example, Algorithm 1 failed to find a descent direction because the mapping is not strongly monotone. But Algorithm 2 converged in 4 iterations for both initial points.

Table 8: Results of Example 4.

		Algorithm 1			Algorithm 2			
CC	Initial point	#Iterations	#f	CPU	#Iterations	#f	#Lemke	CPU
10^{-3}	$(0,\ldots,0)$	17181	failed		4	5	62	0.09
	$(1,\ldots.1)$	16608	failed		4	5	64	0.10
10^{-5}	$(0,\ldots,0)$	17181	failed		4	5	62	0.09
	$(1,\ldots.1)$	16608	failed		4	5	64	0.10

Example 5 This example is the following convex programming problem:

$$\begin{array}{lll} \text{minimize} & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 & \\ & +2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 & \\ \text{subject to} & 4x_1 + 5x_2 - 3x_7 + 9x_8 & \leq 105 \\ & 10x_1 - 8x_2 - 17x_7 + 2x_8 & \leq 0 \\ & -8x_1 + 2x_2 + 5x_9 - 2x_{10} & \leq 12 \\ & 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 & \leq 120 \\ & 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 & \leq 40 \\ & \frac{1}{2}(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 & \leq 30 \\ & x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 & \leq 0 \\ & -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} & \leq 0 \\ & x_i \geq 0, \ i = 1, 2, \ldots, 10, & \end{array}$$

which is formulated as an 18-variable complementarity problem. The results are shown in Table 9. The mapping is monotone but not strongly monotone. Algorithm 1 converged slowly and eventually failed to find a descent direction as the iterate become very close to a solution. On the other hand, Algorithm 2 converged in several iterations for both initial points.

Table 9: Results of Example 5.

CC	Initial point	Algorithm 1 #Iterations	Algorithm 1 #f	Algorithm 1 CPU	Algorithm 2 #Iterations	Algorithm 2 #f	Algorithm 2 #Lemke	Algorithm 2 CPU
10^{-3}	$(0, \ldots, 0)$	31616	414381	71.98	4	5	68	0.14
	$(10, \ldots, 10)$	31238	411704	72.43	6	7	97	0.19
10^{-5}	$(0, \ldots, 0)$	72050	failed		5	6	84	0.17
	$(10, \ldots, 10)$	75257	failed		6	7	97	0.19

Example 6 This example is a spatial price equilibrium problem from Tobin [18] which is formulated as a 42-variable complementarity problem. The mapping is not monotone but is close to be monotone. The results are shown in Table 10.

Table 10: Results of Example 6.

CC	Initial point	Algorithm 1 #Iterations	Algorithm 1 #f	Algorithm 1 CPU	Algorithm 2 #Iterations	Algorithm 2 #f	Algorithm 2 #Lemke	Algorithm 2 CPU
10^{-3}	$(0, \ldots, 0)$	63	148	0.09	6	11	130	1.34
	$(1, \ldots, 1)$	66	155	0.09	7	10	131	1.37
	$(10, \ldots, 10)$	63	149	0.10	7	8	157	1.60
10^{-5}	$(0, \ldots, 0)$	84	199	0.12	7	12	148	1.53
	$(1, \ldots, 1)$	89	209	0.12	7	10	131	1.37
	$(10, \ldots, 10)$	83	197	0.12	7	8	157	1.60

In this example, Algorithms 1 and 2 converged for all initial points chosen in our experiment. Note that the mapping of Example 6 is similar to the form (34) used in the experiments of Section 4, and hence, the mapping is sparse. For the example, Algorithm 1 converged mush faster than Algorithm 2.

Example 7 This example is a traffic assignment problem. This is a 40-variable complementarity problem which is Example 6.2 in Aashtiani [1]. The results are shown in Table 11. The mapping is monotone but not strongly monotone. For this example, Algorithm 1 converged slowly and could not attain the strict convergence criterion CC= 10^{-5}. On the other hand, Algorithm 2 failed because the linear subproblem became unsolvable after 2 or 3 iterations.

Table 11: Results of Example 7.

		Algorithm 1			Algorithm 2			
CC	Initial point	#Iterations	#f	CPU	#Iterations	#f	#Lemke	CPU
10^{-3}	$(0,\dots,0)$	2907	24925	13.25	2		failed	
	$(10,\dots 10)$	2981	25797	13.74	3		failed	
10^{-5}	$(0,\dots,0)$	4218	failed		2		failed	
	$(10,\dots 10)$	4166	failed		3		failed	

6. Concluding remarks

When the mapping is strongly monotone with modulus μ, the solution x^* to (1) satisfies the inequality

$$\| x^* \| \leq \frac{1}{\mu} \| F(0) \| .$$

Hence, we may reform problem (1) as a variational inequality with bounded constraint by adding an extra constraint

$$\| x \|_\infty \leq R$$

with a sufficiently large positive number R, and apply the methods of Fukushima [7] or Taji et al. [17] directly. Then the subproblem becomes a linear variational inequality problem with a bound constraint, which is in general more difficult to solve than a linear complementarity problem of the proposed algorithms.

Since the modulus μ is generally a priori unknown, if the matrix D is chosen not to satisfy $\| D \| < 4\mu$, Algorithm 2 may fail because the search direction is not guaranteed to be a descent direction. When we do not know the exact value of μ for the strongly monotone mapping F, we may start Algorithm 2 with a suitable value of μ, and, if it fails, continue by halving D until convergence is obtained. Eventually we will have $\| D \| < 4\mu$ and hence Algorithm 2 must converge by Theorem 2.

Acknowledgement The authors are grateful to Professor T. Ibaraki of Kyoto University and the refrees for helpful comments.

References

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

[2] Bertsekas, D. P. and Gafni, E. M. : Projection Methods for Variational Inequalities with Application to the Traffic Assignment Problem, *Mathematical Programming Study*, vol. 17 (1982), 139-159.

[3] Cottle, R. W., Pang, J. S. and Stone, R. E.: *The Linear Complementarity Problem*, Academic Press, San Diego, 1992.

[4] Dafermos, S.: Traffic Equilibrium and Variational Inequalities. *Transportation Science*, vol. 14 (1980), 42-54.

[5] Dennis, J. E. and Schnabel, R. B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Eaglewood Cliffs NJ, 1983.

[6] Florian, M.: Mathematical Programming Applications in National, Regional and Urban Planning. *Mathematical Programming: Recent Developments and Applications*, Iri, M. and Tanabe, K. (eds.), KTK Scientific Publishers, Tokyo, 1989.

[7] Fukushima, M.: Equivalent Differentiable Optimization Problems and Descent Methods for Asymmetric Variational Inequality Problems. *Mathematical Programming*, vol. 53 (1992), 99-110.

[8] Garcia, C. B. and Zangwill, W. I.: *Pathways to Solutions, Fixed Points, and Equilibria* Prentice-Hall, Eaglewood Cliffs NJ, 1981.

[9] Harker, P. T.: Accelerating the Convergence of the Diagonalization and Projection Algorithms for Finite-Dimensional Variational Inequalities. *Mathematical Programming*, vol. 41 (1988), 29-59.

[10] Harker, P. T. and Pang, J. S.: Finite-Dimensional Variational Inequality and Nonlinear Complementarity Problems: A Survey of Theory, Algorithms and Applications. *Mathematical Programming*, vol. 48 (1990), 161-220.

[11] Ibaraki, T. and Fukushima, M.: *FORTRAN 77 Optimization Programming* (in Japanese), Iwanami-Shoten, Tokyo, 1991.

[12] Josephy, N. H.: Newton's Method for Generalized Equations. Technical Report No. 1965, Mathematics Research Center, University of Wisconsin, Madison, WI, 1979.

[13] Lemke, C. E.: Bimatrix Equilibrium Points and Mathematical Programming. *Management Science*, vol. 11 (1965), 681-689.

[14] Marcotte, P. and Dussault, J. P.: A Sequential Linear Programming Algorithm for Solving Monotone Variational Inequalities. *SIAM Journal on Control and Optimization*, vol. 27 (1989), 1260-1278.

[15] Mathiesen, L.: An Algorithm Based on a Sequence of Linear Complementarity Problems Applied to a Walrasian Equilibrium Model : An Example. *Mathematical Programming*, vol. 37 (1987), 1-18.

[16] Pang, J. S. and Chan, D.: Iterative Methods for Variational and Complementarity Problems. *Mathematical Programming*, vol. 24 (1982), 284-313.

[17] Taji, K., Fukushima, M. and Ibaraki, T.: A Globally Convergent Newton Method for Solving Strongly Monotone Variational Inequalities. *Mathematical Programming*, vol. 58 (1993), 369-383.

[18] Tobin, R. L.: A Variable Dimension Solution Approach for the General Spatial Price Equilibrium Problem. *Mathematical Programming*, vol. 40 (1988), 33-51.

[19] Van der Laan, G. and Talman, A. J. J.: An Algorithm for the Linear Complementarity Problem with Upper and Lower Bounds. *Journal of Optimization Theory and Applications*, vol. 62 (1989), 151-163.

Kouichi TAJI and Masao FUKUSHIMA:
Graduate School of Information Science,
Advanced Institute of Science and Technology, Nara,
Takayama, Ikoma, Nara 630-01, Japan