Algorithms **2015**, *8*, 1195-1209; doi:10.3390/a8041195

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

A New Smoothing Conjugate Gradient Method for Solving Nonlinear Nonsmooth Complementarity Problems

Ajie Chu, Shouqiang Du * and Yixiao Su

College of Mathematics, Qingdao University, 308 Qingdao Ningxia Road, Qingdao 266071, China;
E-Mails: 13658689660@163.com (A.C.); y_x_su@163.com (Y.S.)

* Author to whom correspondence should be addressed; E-Mail: sqdu@qdu.edu.cn;
Tel.: +86-532-8595-3660.

Academic Editor: Louxin Zhang

Received: 13 October 2015 / Accepted: 11 December 2015 / Published: 17 December 2015

Abstract: In this paper, by using the smoothing Fischer-Burmeister function, we present a new smoothing conjugate gradient method for solving the nonlinear nonsmooth complementarity problems. The line search which we used guarantees the descent of the method. Under suitable conditions, the new smoothing conjugate gradient method is proved globally convergent. Finally, preliminary numerical experiments show that the new method is efficient.

Keywords: smoothing Fischer-Burmeister function; conjugate gradient method; nonlinear nonsmooth complementarity problems

1. Introduction

We consider the nonlinear nonsmooth complementarity problem, which is to find a vector in R^n satisfying the conditions

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

where $F : R^n \rightarrow R^n$ is a locally Lipschitz continuous function. If F is continuously differentiable, then Problem (1) is called the nonlinear complementarity problems NCP(F). As we all know, Equation (1) is a very useful general mathematics model, which is closely related to the mathematical programming, variational inequalities, fixed point problems and mixed strategy problems (such as [1–13]). The methods for solving NCP(F) are classified into three categories: nonsmooth

Newton methods, Jacobian smoothing methods and smoothing methods (see [14–19]). Conjugate gradient methods are widely and increasingly used for solving unconstrained optimization problem, especially in large-scale cases. There are few scholar has investigated the problem how to use the conjugate gradient method to solve NCP(F) (such as [10,20]). Moreover, in these papers, F is required to be a continuous differentiable $P_0 + R_0$ function. In this paper, we present a new smoothing conjugate gradient method for solving Equation (1), where F is only required to be a locally Lipschitz continuous function.

In this paper, we also define the generalized gradient of F at x is

$$\partial F(x) = conv\{\lim_{x_k \longrightarrow x, x_k \in D_F} \bigtriangledown F(x_k)\}$$

where $''conv''$ denotes the convex hull of a set, D_F denotes the set of points at which F is differentiable (see [21]). In the following, we introduce the definition of the smoothing function.

Definition 1 (see [22]) Let $F : R^n \longrightarrow R^n$ be a locally Lipschitz continuous function. We call $\widetilde{F} : R^n \times R_+ \longrightarrow R^n$ is a smoothing function of F, if $\widetilde{F}(x, \mu)$ is continuously differentiable in R^n for any fixed $\mu > 0$, and

$$\lim_{\mu \longrightarrow 0} \widetilde{F}(x, \mu) = F(x)$$

for any fixed $x \in R^n$. If

$$\lim_{x_k \longrightarrow x, \mu \downarrow 0} \bigtriangledown \widetilde{F}(x_k, \mu) \in \partial F(x)$$

for any $x_k \in R^n$, we say F satisfies gradient consistency property.

In the following sections of our paper, we also use the Fischer-Burmeister function (see [23]) and the smoothing Fischer-Burmeister function. (1) The Fischer-Burmeister function

$$\varphi(a, b) = \sqrt{a^2 + b^2} - a - b, (a, b)^T \in R^2$$

where $\varphi : R^2 \longrightarrow R$. From the definition of φ, we know that it is twice continuously differentiable besides $(0, 0)^T$. Moreover, it is a complementarity function, which satisfies

$$\varphi(a, b) = 0 \Longleftrightarrow a \geq 0, b \geq 0, ab = 0$$

Denote

$$H(x) = \begin{pmatrix} \varphi(x_1, F_1(x)) \\ \vdots \\ \varphi(x_n, F_n(x)) \end{pmatrix}$$

It is obvious that $H(x)$ is zero at a point x if and only if x is a solution of Equation (1). Then Equation (1) can be transformed into the following unconstrained optimization problem

$$\min \Psi(x) = \frac{1}{2}\|H(x)\|^2$$

We know that the optimal value of Ψ is zero, and Ψ is called the value function of Equation (1).

(2) The smoothing Fischer-Burmeister function

$$\varphi_\mu(a, b) = \sqrt{a^2 + b^2 + \mu} - a - b$$

where $\varphi : R^3 \longrightarrow R$ and $\mu > 0$.

Let

$$H_\mu(x) = \begin{pmatrix} \varphi_\mu(x_1, \tilde{F}_1(x,\mu)) \\ \vdots \\ \varphi_\mu(x_n, \tilde{F}_n(x,\mu)) \end{pmatrix}$$

$$\Psi_\mu(x) = \frac{1}{2}\|H_\mu(x)\|^2$$

where $\tilde{F}_i(x,\mu)$ is the smoothing function of $F_i(x), i = 1, \cdots, n$.

The rest of this work is organized as follows. In Section 2, we describe the new smoothing conjugate gradient method for the solution of Problem (1). It is shown that this method has global convergence properties under fairly mild assumptions. In Section 3, preliminary numerical results and some discussions for this method are presented.

2. The New Smoothing Conjugate Gradient Method and its Global Convergence

The new smoothing conjugate gradient direction is defined as

$$d_k = \begin{cases} -\nabla\Psi_{\mu_0}(x_0), & k = 0 \\ -\nabla\Psi_{\mu_{k-1}}(x_k) + \beta_k d_{k-1}, & k \geq 1 \end{cases} \tag{2}$$

where β_k is a scalar. Here, we use β_k (see [24]) which is defined as

$$\beta_k^{DY} = \frac{\|\nabla\Psi_{\mu_{k-1}}(x_k)\|^2}{d_{k-1}^T y_{k-1}} \tag{3}$$

where $y_{k-1} = \nabla\Psi_{\mu_{k-1}}(x_k) - \nabla\Psi_{\mu_{k-2}}(x_{k-1})$. When $k = 1$, we set $\mu_{k-2} = \mu_0$. The line search is Armijo-type line search (see [25]), where $\alpha_k = \eta^{m_k}, 0 < \eta < 1, m_k$ is the smallest nonnegative integer satisfies

$$\Psi_{\mu_k}(x_k + \alpha_k d_k) \leq \Psi_{\mu_k}(x_k) + \delta\alpha_k(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k \tag{4}$$

$$(\nabla\Psi_{\mu_k}(x_k + \alpha_k d_k))^T d_{k+1} \leq -\sigma\|\nabla\Psi_{\mu_k}(x_k + \alpha_k d_k)\|^2, 0 < \sigma \leq 1, 0 < \delta < 1 \tag{5}$$

Then, we give the new smoothing conjugate gradient method for solving Equation (1).

Algorithm 1: Smoothing Conjugate Gradient Method

(S.0) Choose $x_0 \in R^n, \varepsilon > 0, \mu_0 > 0, m > 0, \sigma, \delta, m_1 \in (0,1), d_0 = -\nabla\Psi_{\mu_0}(x_0)$. Set $k = 0$.

(S.1) If $\Psi(x_k) \leq \varepsilon$, then stop, otherwise go to Step 2.

(S.2) Compute α_k by Equations (4) and (5), where $d_{k+1} = -\nabla\Psi_{\mu_k}(x_k + \alpha_k d_k) + \beta_{k+1} d_k$ and β_{k+1} is given by Equation (3). Let $x_{k+1} = x_k + \alpha_k d_k$.

(S.3) If $\|\nabla\Psi_{\mu_k}(x_{k+1})\| \geq m\mu_k$, then set $\mu_{k+1} = \mu_k$, otherwise set $\mu_{k+1} = m_1\mu_k$.

(S.4) Let $k = k + 1$, go back to Step 1.

Algorithm 2: Algorithm Framework of Algorithm 1

```
PROGRAM ALGORITHM
    INITIALIZE x_0 ∈ R^n, ε > 0, μ_0 > 0, m > 0, m_1 ∈ (0,1);
    Set k = 0 and d_0 = −∇Ψ_{μ_0}(x_0).
    WHILE the termination condition
            Ψ(x_k) ≤ ε is not met
        Find step sizes α_k;
        Set x_{k+1} = x_k + α_k d_k
        Evaluate ∇Ψ_{μ_k}(x_{k+1}) and d_{k+1};
        IF ||∇Ψ_{μ_k}(x_{k+1})|| ≥ m μ_k THEN
            Set μ_{k+1} = μ_k;
        ELSE
            Set μ_{k+1} = m_1 μ_k;
        END IF
        Set k = k + 1;
    END WHILE
    RETURN final solution x_k;
END ALGORITHM
```

In the following, we will give the analysis about the global convergence of Algorithm 1. (The Algorithm 2 is the algorithm framework of Algorithm 1.) Before doing this work, we need the following basic assumptions.

Assumption 1.

(i) For any $x \in R^n$, $0 < \mu \leq \mu_0$, the level set $L_\mu(c) = \{x \in R^n | \Psi_\mu(x) \leq c\}$ is bounded.
(ii) $\nabla\Psi_{\mu_k}(x_{k+1})$ is Lipschitz continuous, that is, there exists a constant $L > 0$ such that

$$\|\nabla\Psi_{\mu_k}(x_{k+1}) - \nabla\Psi_{\mu_{k-1}}(x_k)\| \leq L\|x_{k+1} - x_k\|, \forall x_{k+1}, x_k \in L_\mu(c)$$

Lemma 1. Suppose that $\{d_k\}$ is an infinite sequence of directions generated by Algorithm 1, then

$$-(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k \geq \bar{c}\|\nabla\Psi_{\mu_{k-1}}(x_k)\|^2, \forall k \geq 0, \bar{c} < \sigma \tag{6}$$

Proof If $k = 0$, by Equation (2) and $\bar{c} < 1$, we can know that Equation (6) holds. If $k > 0$, by Equation (5) and $\bar{c} < \sigma$, we can conclude that Equation (6) holds.

Lemma 2. Suppose that Assumption 1 holds. Then, there exists $\alpha_k > 0$ for every k, and

$$\alpha_k \geq \omega \frac{|(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k|}{\|d_k\|^2} \tag{7}$$

with ω is a positive constant.

Proof From Step 0 of Algorithm 1, we know that $d_0 = -\nabla\Psi_{\mu_0}(x_0)$, *i.e.*, d_0 is a descent direction. Suppose that d_k is satisfied

$$(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k \leq -\sigma\|\nabla\Psi_{\mu_{k-1}}(x_k)\|^2 \leq 0 \tag{8}$$

for any $\tilde{\alpha}_k$. We denote

$$\tilde{x}_{k+1} = x_k + \tilde{\alpha}_k d_k$$
$$\tilde{d}_{k+1} = -\nabla\Psi_{\mu_k}(\tilde{x}_{k+1}) + \tilde{\beta}_{k+1} d_k \tag{9}$$

By

$$(\nabla\Psi_{\mu_k}(\tilde{x}_{k+1}))^T \tilde{d}_{k+1} = -\|\nabla\Psi_{\mu_k}(\tilde{x}_{k+1})\|^2 + \tilde{\beta}_{k+1}[(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k + (\nabla\Psi_{\mu_k}(\tilde{x}_{k+1}) - \nabla\Psi_{\mu_{k-1}}(x_k))^T d_k] \tag{10}$$

We know that β_k in Equation (3) is equivalent to (see [24])

$$\beta_k = \frac{(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k}{(\nabla\Psi_{\mu_{k-2}}(x_{k-1}))^T d_{k-1}} > 0 \tag{11}$$

Since Assumption 1, Equations (10) and (11) yield

$$(\nabla\Psi_{\mu_k}(\tilde{x}_{k+1}))^T \tilde{d}_{k+1} \le -\sigma\|\nabla\Psi_{\mu_k}(\tilde{x}_{k+1})\|^2, \tilde{\alpha}_k \in (0, \frac{|(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k|}{L\|d_k\|^2}) \tag{12}$$

by Mean Value Theorem, we have

$$\Psi_{\mu_k}(\tilde{x}_{k+1}) - \Psi_{\mu_k}(x_k) = \int_0^1 \tilde{\alpha}_k(\nabla\Psi_{\mu_k}(x_k + t\tilde{\alpha}_k d_k))^T d_k dt$$
$$= \tilde{\alpha}_k(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k + \int_0^1 \tilde{\alpha}_k[\nabla\Psi_{\mu_k}(x_k + t\tilde{\alpha}_k d_k) - \nabla\Psi_{\mu_{k-1}}(x_k)]^T d_k dt$$
$$\le \tilde{\alpha}_k(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k + \int_0^1 L\tilde{\alpha}_k^2\|d_k\|^2 t dt$$
$$\le \tilde{\alpha}_k(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k + \frac{1}{2}L\tilde{\alpha}_k^2\|d_k\|^2$$

Then, we obtain that

$$\Psi_{\mu_k}(\tilde{x}_{k+1}) - \Psi_{\mu_k}(x_k) \le \delta\tilde{\alpha}_k(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k,$$
$$\forall\tilde{\alpha}_k \in (0, \frac{2(1-\delta)}{L}\frac{|(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k|}{\|d_k\|^2}) \tag{13}$$

By Equations (12) and (13), we know that Equations (4) and (5) determine a positive stepsize α_k. And there must exists a constant $\xi \in (0,1)$ yields

$$\xi \cdot \frac{|(\nabla\Psi_{\mu_{k-1}}(x_k))^T d_k|}{L\|d_k\|^2} < 1$$

Denote $\omega = \min\{\frac{\xi}{L}, \frac{2\xi(1-\delta)}{L}\}$, then Equation (7) holds. And Equation (5) implies that Equation (8) holds for $k+1$. Hence, the proof is completed.

Theorem 1. Suppose that for any fixed $\mu > 0$, Ψ_μ satisfies Assumption 1, then the infinite sequence $\{x_k\}$ generated by Algorithm 1 satisfies

$$\lim_{k\longrightarrow\infty} \mu_k = 0, \ \liminf_{k\longrightarrow\infty} \|\nabla\Psi_{\mu_{k-1}}(x_k)\| = 0 \tag{14}$$

Proof Denote $K = \{k|\mu_{k+1} = m_1\mu_k\}$, we first show that K is an infinite set.
If K is a finite set, there exists an integer $\bar{k}$ such that

$$\|\nabla\Psi_{\mu_{k-1}}(x_k)\| \geq m\mu_{k-1}$$

for all $k > \bar{k}$. We also have $\mu_k = \mu_{\bar{k}} =: \bar{\mu}$ for all $k > \bar{k}$ and

$$\liminf_{k\longrightarrow\infty} \|\nabla\Psi_{\bar{\mu}}(x_k)\| > 0 \tag{15}$$

In the following, we will proof

$$\liminf_{k\longrightarrow\infty} \|\nabla\Psi_{\bar{\mu}}(x_k)\| = 0 \tag{16}$$

By Lemma 1 and Assumption 1, we know that $\{\Psi_{\bar{\mu}}(x_k)\}$ is a monotone decreasing sequence and the limit of $\{\Psi_{\bar{\mu}}(x_k)\}$ is exist. Summing Equation (7), we get

$$\sum_{k\geq\bar{k}+1} \frac{(\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2}{\|d_k\|^2} < \infty \tag{17}$$

Due to Equation (2), we also have

$$d_k + \nabla\Psi_{\bar{\mu}}(x_k) = \beta_k d_{k-1}, \forall k \geq \bar{k} + 1 \tag{18}$$

Square both sides of Equation (18), we get

$$\|d_k\|^2 = (\beta_k)^2\|d_{k-1}\|^2 - 2(\nabla\Psi_{\bar{\mu}}(x_k))^T d_k - \|\nabla\Psi_{\bar{\mu}}(x_k)\|^2 \tag{19}$$

Divided both sides of Equation (19) by $((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2$, we have

$$\begin{aligned}
\frac{\|d_k\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} &= \frac{(\beta_k)^2\|d_{k-1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} - \frac{2(\nabla\Psi_{\bar{\mu}}(x_k))^T d_k}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} - \frac{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} \\
&= \frac{(\beta_k)^2\|d_{k-1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} - (\frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|} + \frac{\|\nabla\Psi_{\bar{\mu}}(x_k)\|}{(\nabla\Psi_{\bar{\mu}}(x_k))^T d_k})^2 + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2} \\
&\leq \frac{(\beta_k)^2\|d_{k-1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2} \\
&= \frac{\|d_{k-1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_{k-1}))^T d_{k-1})^2} + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2} \\
&\leq \frac{\|d_{k-2}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_{k-2}))^T d_{k-2})^2} + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_{k-1})\|^2} + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2} \\
&\leq \cdots \\
&\leq \frac{\|d_{\bar{k}+1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_{\bar{k}+1}))^T d_{\bar{k}+1})^2} + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_{\bar{k}+2})\|^2} + \cdots + \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_k)\|^2}
\end{aligned}$$

Denote

$$\frac{\|d_{\bar{k}+1}\|^2}{((\nabla\Psi_{\bar{\mu}}(x_{\bar{k}+1}))^T d_{\bar{k}+1})^2} = \lambda > 0$$

Then

$$\frac{\|d_k\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} \leq \lambda + \sum_{i=\bar{k}+2}^{k} \frac{1}{\|\nabla\Psi_{\bar{\mu}}(x_i)\|^2} \tag{20}$$

If Equation (16) is not hold, there exists $\gamma > 0$ such that

$$\|\nabla\Psi_{\bar{\mu}}(x_k)\| \geq \gamma, \forall k > \bar{k} + 1 \tag{21}$$

We obtain from Equations (20) and (21) that

$$\frac{\|d_k\|^2}{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2} \leq \frac{\lambda\gamma^2 + k - \bar{k} - 1}{\gamma^2}$$

Because of

$$\frac{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2}{\|d_k\|^2} \geq \frac{\gamma^2}{\lambda\gamma^2 + k - \bar{k} - 1}$$

provies

$$\sum_{k \geq \bar{k}+1} \frac{((\nabla\Psi_{\bar{\mu}}(x_k))^T d_k)^2}{\|d_k\|^2} = +\infty$$

which leads to a contradiction with Equation (17). This show that Equation (16) holds. There are conflicts between Equations (16) and (15). This show that K must be an infinite set and

$$\lim_{k \longrightarrow \infty} \mu_k = 0 \tag{22}$$

Then, we can assume that $K = \{k_0, k_1, ...\}$ with $k_0 < k_1 < ...$ Hence, we get

$$\lim_{i \longrightarrow \infty} \|\nabla\Psi_{\mu_{k_i}}(x_{k_i+1})\| \leq m \lim_{i \longrightarrow \infty} \mu_{k_i} = 0$$

and completes the proof.

3. Numerical Tests

In this section, we intend to test the efficiency of Algorithm 1 by numerical experiments. We use Algorithm 1 to solve eleven examples, some of them are proposed the first time, some of them are modified by the examples of the references (such as [26,27]).

The smoothing function of F as $\tilde{F}_i(x, \mu) = \sqrt{F_i(x)^2 + \mu}$ is used in solving Examples 1–4. From Example 5 to Example 11, the smoothing function of F is defined by (see [26]). $\tilde{F}(x, \mu) = \mu \ln \sum_{i=1}^{m} \exp(\frac{f_i(x)}{\mu})$, where $F(x) = \max\{f_1(x), \cdots, f_m(x)\}$, $i = 1, \cdots m$.

Throughout the experiments, we set $\sigma = 10^{-2}, m = 1.5, m_1 = 0.5$. In Examples 1–3 and Examples 5–8, we set $\varepsilon = 10^{-4}, \delta = 10^{-3}, \eta = 0.4, \mu_0 = 0.2$. Example 4, in which we set parameters $\varepsilon = 10^{-3}, \delta = 10^{-2}, \eta = 0.1, \mu_0 = 0.02$. In the case of Examples 9–11, we set $\varepsilon = 10^{-2}, \delta = 10^{-3}, \eta = 0.4, \mu_0 = 0.2$. We choose $\Psi(x) \leq \varepsilon$ as the termination criterion. Our numerical results are summaried in Tables 1–11, where all components of x_0 are randomly selected from 0 to 10. We randomly generate 10 initial points, then implement Algorithm 1 to solve the test problem for each initial point. By the numerical of results of Examples 10–11, we can see that Algorithm 1 is suitably to solve the large scale problems.

Example 1. We consider Equation (1), where F is defined by $F(x) = |2x - 1|$.

The exact solutions of this problem are 0 and 0.5.

Table 1. Number of iterations and the final value of $\Psi(x^*)$.

$\boldsymbol{x_0}$	$\boldsymbol{x^*}$	$\boldsymbol{\Psi(x^*)}$	$\boldsymbol{k}$
0.9713	5.053658e−1	5.636783e−5	1
1.7119	4.977618e−1	9.929266e−6	11
2.7850	−1.295343e−2	8.495830e−5	8
3.1710	5.422178e−3	1.461954e−5	8
4.0014	5.562478e−3	1.538368e−5	8
5.4688	−7.521520e−3	2.849662e−5	7
6.5574	5.926470e−3	1.745635e−5	10
7.9221	1.276205e−2	8.037197e−5	7
8.4913	−1.994188e−3	1.992344e−6	7
9.3399	1.723553e−3	1.482749e−6	7

Example 2. We consider Equation (1), where $F = \begin{pmatrix} |2x_1 - 1| \\ |4x_2 + x_1 - \frac{1}{2}| \end{pmatrix}$. There are three exact solutions as $(\frac{1}{2}, 0)^T$, $(0, \frac{1}{8})^T$ and $(0, 0)^T$.

Table 2. Number of iterations and the final value of $\Psi(x^*)$.

$\boldsymbol{x_0}$	$\boldsymbol{x^*}$	$\boldsymbol{\Psi(x^*)}$	$\boldsymbol{k}$
$(4.6939, 0.1190)^T$	$(-0.0058, 0.0031)^T$	2.162427e−5	7
$(5.2853, 1.6565)^T$	$(0.0077, 0.1233)^T$	2.974655e−5	13
$(9.9613, 0.7818)^T$	$(0.0030, 0.1248)^T$	7.106107e−6	5
$(4.9836, 9.5974))^T$	$(0.4979, -0.0097)^T$	6.744680e−5	12
$(1.4495, 8.5303)^T$	$(0.4978, 0.0080)^T$	3.327241e−5	13
$(0.4965, 9.0272)^T$	$(0.0045, 0.1221)^T$	3.348105e−5	15
$(9.1065, 1.8185)^T$	$(-0.0045, 0.1296)^T$	9.857281e−5	6
$(4.0391, 0.9645)^T$	$(0.0087, 0.1247)^T$	6.417282e−5	10
$(7.7571, 4.8679)^T$	$(0.0045, -0.0043)^T$	1.946526e−5	13
$(7.0605, 0.3183)^T$	$(0.0086, 0.122)^T$	4.037476e−5	8

Example 3. We consider Equation (1), where $F = \begin{pmatrix} |5x_1 + x_2 - x_3| \\ x_1^2 + 4x_2 - x_3 - 2 \\ 5x_2^2 - 6x_1 - 2x_3 \end{pmatrix}$.

$(0, \frac{1}{2}, 0)^T$ is one of the exact solutions of this problem.

Table 3. Number of iterations and the final value of $\Psi(x^*)$.

$\boldsymbol{x_0}$	$\boldsymbol{x^*}$	$\boldsymbol{\Psi(x^*)}$	$\boldsymbol{k}$
$(1.9175, 7.3843, 2.4285)^T$	$(0.0087, 0.5021, -0.0026)^T$	9.785244e−5	21
$(1.1921, 9.3983, 6.4555)^T$	$(0.0047, 0.5019, 0.0102)^T$	6.577107e−5	25
$(1.8687, 4.8976, 4.4559)^T$	$(-0.0055, 0.4974, -0.0109)^T$	7.552798e−5	19
$(2.7029, 2.0846, 5.6498)^T$	$(0.0099, 0.4998, 0.0028)^T$	5.759549e−5	26
$(7.2866, 7.3784, 0.6340)^T$	$(0.0099, 0.5003, -0.0063)^T$	9.612693e−5	36
$(1.2991, 5.6882, 4.6939)^T$	$(-0.0115, 0.5009, 0.0065)^T$	9.216436e−5	31
$(5.3834, 9.9613, 0.7818)^T$	$(0.0022, 0.4952, -0.0083)^T$	9.798255e−5	26
$(9.5613, 5.7521, 0.5978)^T$	$(0.0089, 0.5029, 0.0074)^T$	7.570081e−5	28
$(7.7571, 4.8679, 4.3586)^T$	$(0.0048, 0.5025, -0.0008)^T$	6.774521e−5	24
$(3.8827, 5.5178, 2.2895)^T$	$(-0.0047, 0.4981, -0.0111)^T$	7.903206e-5	25

Example 4. We consider Equation (1), where $F = \begin{pmatrix} |2x_1 - x_2 + 3x_3 + 2x_4 - 6| \\ 3x_1 - 3x_2 + 3x_3 + 2x_4 - 5 \\ 3x_1 - x_2 - x_3 + 2x_4 - 3 \\ 3x_1 - x_2 + 3x_3 - x_4 - 4 \end{pmatrix}$.
$(\frac{31}{13}, \frac{22}{13}, 0, \frac{19}{13})^T$, $(\frac{7}{4}, 0, 0, \frac{5}{4})^T$, $(0, 0, \frac{11}{5}, \frac{13}{5})^T$, $(3, 0, 0, 0)^T$ are four of the exact solutions of this problem.

Table 4. Number of iterations and the final value of $\Psi(x^*)$.

$\boldsymbol{x_0}$	$\boldsymbol{x^*}$	$\boldsymbol{\Psi(x^*)}$	$\boldsymbol{k}$
$(5.6743, 9.6878, 8.2450, 9.5961)^T$	$(0.0062, -0.0249, 2.1692, 2.5652)^T$	4.436641e-4	21
$(0.1485, 1.5669, 4.7157, 5.4299)^T$	$(0.0009, 0.0320, 2.2173, 2.6291)^T$	5.987181e−4	37
$(0.5969, 6.5803, 8.8964, 1.0963)^T$	$(3.0522, 0.0031, -0.0202, -0.0151)^T$	3.790585e−4	23
$(8.7494, 1.2100, 8.5635, 8.9978)^T$	$(0.0296, -0.0290, 2.1300, 2.5000)^T$	9.638295e−4	17
$(7.7836, 0.6937, 2.7878, 3.7937)^T$	$(3.0003, 0.0096, 0.0154, -0.0158)^T$	3.052065e−4	13
$(0.6837, 0.8497, 0.6834, 4.0982)^T$	$(3.0055, 0.0333, -0.0050, 0.0131)^T$	7.098848e−4	21
$(7.6034, 5.8410, 4.0295, 5.1004)^T$	$(2.4048, 1.7076, -0.0096, 1.4753)^T$	4.242339e−4	25
$(9.8754, 9.2271, 5.6426, 4.3146)^T$	$(3.0166, 0.0148, -0.0271, 0.0280)^T$	8.913504e−4	20
$(8.5061, 1.4453, 3.7049, 6.2239)^T$	$(2.9635, 0.0040, 0.0176, 0.0220)^T$	5.981987e−4	26
$(2.7744, 0.0611, 3.7471, 4.3693)^T$	$(0.0132, -0.0399, 2.1529, 2.5310)^T$	9.794091e−4	21

Example 5. We consider Equation (1), where $F = max\{(x - 2), (2x - 5)\}$. There are two exact solutions as 0 and 2.

Table 5. Number of iterations and the final value of $\Psi(x^*)$.

x_0	x^*	$\Psi(x^*)$	k
0.2922	2.0024	2.787626e−6	5
1.7071	1.9894	5.621467e−5	3
2.2766	2.0075	2.836408e−5	3
3.1110	2.0001	2.938429e−9	1
4.3570	2.0101	5.061011e−5	4
5.7853	2.0109	5.937701e−5	5
6.2406	1.9871	8.325445e−5	6
7.1122	2.0116	6.635145e−5	3
8.8517	1.9970	4.557770e−6	6
9.7975	1.9928	2.607803e−5	4

Example 6. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \end{pmatrix}$.
$f_i(x) = max\{x_1^2, x_2^2, x_3^2, x_4^2\}$, $i = 1, 2, 3, 4$. The exact solution of this problem is $(0, 0, 0, 0)^T$.

Table 6. Number of iterations and the final value of $\Psi(x^*)$.

x_0	x^*	$\Psi(x^*)$	k
$(7.4003, 2.3483, 7.3496, 9.7060)^T$	$(0.0825, 0.0842, 0.0825, 0.0830)^T$	9.228501e−5	29
$(1.3393, 0.3089, 9.3914, 3.0131)^T$	$(0.0858, 0.0629, 0.0484, 0.0414)^T$	9.446089e−5	22
$(7.3434, 0.5133, 0.7289, 0.8853)^T$	$(0.0852, 0.0817, 0.0801, 0.0607)^T$	9.546345e−5	36
$(6.7865, 4.9518, 1.8971, 4.9501)^T$	$(0.0431, 0.0800, 0.0733, 0.0801)^T$	7.428734e−5	34
$(1.4761, 0.5497, 8.5071, 5.6056)^T$	$(0.0774, 0.0691, 0.0717, 0.0655)^T$	6.591548e−5	39
$(0.5670, 5.2189, 3.3585, 1.7567)^T$	$(0.0244, 0.0421, 0.0575, 0.0604)^T$	2.433055e−5	25
$(7.6903, 5.8145, 9.2831, 5.8009)^T$	$(0.0477, 0.0739, 0.0745, 0.0766)^T$	6.280631e−5	32
$(6.9475, 7.5810, 4.3264, 6.5550)^T$	$(0.0846, 0.0819, 0.0566, 0.0850)^T$	9.476538e−5	21
$(2.8785, 4.1452, 4.6484, 7.6396)^T$	$(0.0637, 0.0802, 0.0742, 0.0012)^T$	5.739573e−5	33
$(2.9735, 0.6205, 2.9824, 0.4635)^T$	$(0.0854, 0.0727, 0.0831, 0.0542)^T$	9.575717e−5	22

Example 7. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ \vdots \\ f_{10}(x) \end{pmatrix}$
with $f_i(x) = max\{x_1^2, \cdots, x_{10}^2\}$, $i = 1, \cdots, 10$. The exact solution of this problem is $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$.

Table 7. Number of iterations and the final value of $\Psi(x^*)$.

x_0	$\Psi(x^*)$	k
$(8.2408, 8.2798, 2.9337, 3.0937, 5.2303, 3.2530, 8.3184, 8.1029, 5.5700, 2.6296)^T$	9.719070e−5	27
$(9.5089, 4.4396, 0.6002, 8.6675, 6.3119, 3.5507, 9.9700, 2.2417, 6.5245, 6.0499)^T$	9.957464e−5	45
$(4.1705, 9.7179, 9.8797, 8.6415, 3.8888, 4.5474, 2.4669, 7.8442, 8.8284, 9.1371)^T$	8.965459e−5	39
$(8.3975, 3.7172, 8.2822, 1.7652, 1.2952, 8.7988, 0.4408, 6.8672, 7.3377, 4.3717)^T$	9.644608e−5	47
$(9.7209, 0.3146, 8.3540, 8.3571, 0.4986, 5.4589, 9.4317, 3.2147, 8.0647, 6.0140)^T$	9.737485e−5	37
$(8.3336, 4.0363, 3.9018, 3.6045, 1.4026, 2.6013, 0.8682, 4.2940, 2.5728, 2.9756)^T$	9.240212e−5	47
$(4.8267, 3.7601, 5.2378, 2.6487, 0.6836, 4.3633, 1.7385, 0.2611, 9.5468, 4.3060)^T$	8.801643e−5	44
$(0.5398, 0.2062, 6.8148, 5.9863, 1.1403, 7.9625, 6.1785, 0.7021, 0.6928, 1.3601)^T$	8.946151e−5	44
$(5.7099, 1.6977, 1.4766, 4.7608, 9.0810, 5.5218, 0.3294, 0.5386, 8.0506, 4.5137)^T$	8.815143e−5	45
$(2.1647, 7.8620, 7.2309, 2.7884, 5.8243, 4.2101, 0.9207, 0.2403, 4.9115, 2.7827)^T$	6.697806e−5	39

Example 8. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ \vdots \\ f_4(x) \end{pmatrix}$ with $f_i(x) = \sum_{j=1}^{4} max\{-x_j - x_{j+1}, -x_j - x_{j+1} + (x_j^2 + x_{j+1}^2 + 1)\}$, $i = 1, 2, 3, 4$. The exact solution of this problem is $(0, 0, 0, 0)^T$.

Table 8. Number of iterations and the final value of $\Psi(x^*)$.

x_0	$\Psi(x^*)$	k
$(4.1131, 8.2898, 9.3511, 3.9907)^T$	3.670149e−5	4
$(0.5221, 5.7119, 7.4767, 3.2024)^T$	4.216994e−5	4
$(5.4000, 2.2106, 0.9595, 0.6017)^T$	6.167554e−5	7
$(6.6015, 0.5231, 5.5683, 7.1203)^T$	3.838925e−5	4
$(1.6924, 2.5845, 1.9791, 6.0569)^T$	6.272257e−5	6
$(3.3969, 1.9786, 5.0683, 9.5076)^T$	7.097729e−5	5
$(4.2175, 4.1131, 9.5914, 7.5025)^T$	2.693701e−5	4
$(8.8728, 0.5585, 1.3822, 8.6306)^T$	9.021922e−5	7
$(9.8100, 2.3352, 0.9623, 3.8458)^T$	4.687797e−5	5
$(9.6426, 6.7115, 2.9917, 5.3113)^T$	8.657057e−5	6

Example 9. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ \vdots \\ f_4(x) \end{pmatrix}$ with $f_i(x) = \sum_{j=1}^{4} max\{-x_j - x_{j+1}, -x_j - x_{j+1} + (x_j^2 + x_{j+1}^2 - 1)\}$, $i = 1, 2, 3, 4$. The exact solution of this problem is $(0, 0, 0, 0)^T$.

Table 9. Number of iterations and the final value of $\Psi(x^*)$.

x_0	$\Psi(x^*)$	k
$(1.5290, 1.5254, 1.5555, 0.8957)^T$	9.777886e−3	8
$(4.5442, 6.6890, 8.3130, 7.9024)^T$	3.912481e−3	5
$(9.0150, 3.1834, 5.9708, 2.9780)^T$	9.081688e−3	3
$(3.1781, 9.8445, 5.4825, 7.4925)^T$	6.868711e−3	7
$(8.4185, 1.6689, 9.0310, 1.0512)^T$	5.318627e−3	4
$(7.4509, 7.2937, 7.1747, 1.3343)^T$	7.203761e−3	9
$(4.4579, 5.0879, 5.3049, 8.5972)^T$	9.500345e−3	4
$(6.7772, 8.0584, 5.3124, 9.5590)^T$	9.421194e−3	4
$(0.6668, 5.4152, 2.8166, 4.8090)^T$	6.718722e−3	7
$(6.8486, 2.0826, 6.0816, 3.2618)^T$	3.494877e−3	4

Example 10. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix}$ with $f_i(x) = max\{x_1^2 - 6x_1, \cdots, x_n^2 - 6x_n\}$, $i = 1, \cdots, n$. n represents the problem dimension. The solution is $x^* = (\lambda \cdots \lambda)^T$ (λ is no more than 6). In this problem, we intend to check the efficiency of Algorithm 1 with the dimension of test problem is $50, 100$, and 200. We randomly selected ten initial values when $n = 50$, $n = 100$ and $n = 200$.

Table 10. Number of iterations, the final value of $\Psi(x^*)$ and dimension of the test problem.

n = 50		n = 100		n = 200	
$\Psi(x^*)$	k	$\Psi(x^*)$	k	$\Psi(x^*)$	k
1.625691e−3	9	9.444914e−3	11	9.897292e−3	15
4.082584e−3	7	5.358975e−5	9	3.937758e−4	5
6.082289e−3	7	4.734809e−3	9	5.800944e−3	16
2.042082e−3	9	3.249863e−3	6	3.289200e−3	11
3.765484e−3	9	6.587880e−3	10	4.674659e−3	10
7.553578e−3	13	2.632872e−3	10	1.450852e−3	13
4.208302e−4	14	4.177174e−3	3	9.461359e−3	16
4.250316e−3	9	9.744427e−3	7	3.778464e−3	15
2.634965e−5	10	5.854241e−6	10	1.501579e−3	8
3.445498e−3	11	4.209193e−3	6	1.984871e−3	25

Example 11. We consider Equation (1), where $F = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix}$ with $f_i(x) = max\{x_1^2, \cdots, x_n^2\}$, $i = 1, \cdots, n$. The problem has only unique solution $x^* = (0, \cdots, 0)^T$. We randomly selected ten initial values when $n = 100$, $n = 200$ and $n = 500$.

Table 11. Number of iterations, the final value of $\Psi(x^*)$ and dimension of the test problem.

n = 100		**n = 200**		**n = 500**	
$\Psi(x^*)$	k	$\Psi(x^*)$	k	$\Psi(x^*)$	k
9.152621e−03	17	9.040255e−03	9	7.682471e−3	14
4.383679e−3	15	6.976857e−3	9	8.861191e−3	15
5.172738e−3	15	6.902897e−3	10	8.892858e−3	12
5.796109e−3	12	7.686345e−3	12	9.210427e−3	14
7.613768e−3	16	8.400876e−3	10	9.843579e−3	10
5.398565e−3	12	8.066523e−3	10	9.717126e−3	13
3.403516e−3	15	9.097423e−3	12	8.999900e−3	15
8.701785e−3	13	7.208014e−3	11	9.970099e−3	12
8.302172e−3	11	7.822304e−3	13	9.391355e−3	15
6.610621e−3	13	7.278306e-3	9	9.624919e−3	10

4. Conclusions

In this paper, we have presented a new smoothing conjugate gradient method for the nonlinear nonsmooth complementarity problems. The method is based on a smoothing Fischer-Burmeister function and Armijo-type line search. With careful analysis, we are able to show that our method is globally convergent. Numerical tests illustrate that the method can efficiently solve the given test problems, therefor the new method is promising. We might consider more effective ways of choosing smoothing functions and line search methods for our method. This remains under investigation.

Acknowledgments

The authors wish to thank the anonymous referees for their helpful comments and suggestions, which led to great improvement of the paper. This work is also supported by National Natural Science Foundation of China (NO. 11101231, 11401331), Natural Science Foundation of Shandong (No. ZR2015AQ013) and Key Issues of Statistical Research of Shandong Province (KT15173).

Author Contributions

Ajie Chu prepared the manuscript. Yixiao Su assisted in the work. Shouqiang Du was in charge of the overall research of the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Facchinei, F.; Pang, J.S. *Finite-Demensional Variational Inequalities and Complementarity Problems*; Spring-Verlag: New York, NY, USA, 2003.
2. Luca, T.D.; Facchinei, F.; Kanzow, C. A semismooth equation approach to the solution of nonlinear complementarity problems. *Math. Programm.* **1996**, *75*, 407–439.
3. Ferris, M.C.; Pang, J.S. Engineering and economic applications of complementarity problems. *SIAM Rev.* **1997**, *39*, 669–713.
4. Zhao, Y.B.; Li, D. A new path-following algorithm for nonlinear complementarity problems. *Comp. Optim. Appl.* **2005**, *34*, 183–214.
5. Yu, Q.; Huang, C.C.; Wang, X.J. A combined homotopy interior point method for the linear complementarity problem. *Appl. Math. Comp.* **2006**, *179*, 696–701.
6. Wang, Y.; Zhao, J.X. An algorithm for a class of nonlinear complementarity problems with non-Lipschitzianfunctions. *Appl. Numer. Math.* **2014**, *82*, 68–79.
7. Fischer, A.; Jiang, H. Merit functions for complementarity and related problems: A survey. *Comp. Optim. Appl.* **2000**, *17*, 159–182.
8. Chen, J.S.; Pan, S.H. A family of NCP functions and a descent method for the nonlinear complementarity problem. *Comp. Optim. Appl.* **2008**, *40*, 389–404.
9. Luca, T.D.; Facchinei, F.; Kanzow, C. A theoretical and numerical comparison of some semismooth algorithm for complementarity problems. *Comp. Optim. Appl.* **2000**, *16*, 173–205.
10. Wu, C.Y. *The Conjugate Gradient Method for Solving Nonlinear Complementarity Problems*; Inner Mongolia University: Hohhot, China, 2012.
11. Qi, L.; Sun, D. Nonsmooth and smoothing methods for nonlinear complementarity problems and variational inequalities. *Encycl. Optim.* **2009**, *1*, 2671–2675.
12. Facchinei, F.; Kanzow, C. A nonsmooth inexact Newton method for the solution of large-scale nonlinear complementarity problems. *Math. Programm.* **1997**, *76*, 493–512.
13. Yang, Y.F.; Qi, L. Smoothing trust region methods for nonlinear complementarity problems with P_0 -functions. *Ann. Op. Res.* **2005**, *133*, 99–117.
14. Chen, B.; Xiu, N. A global linear and local quadratic non-interior continuation method for nonlinear complementarity problems based on Chen-Mangasarian smoothing functions. *SIAM J. Optim.* **1999**, *9*, 605–623.
15. Chen, B.; Chen, X.; Kanzow, C. A penalized Fischer-Burmeister NCP-function: Theoretical investigation and numerical results. *Math. Programm.* **2000**, *88*, 211–216.
16. Kanzow, C.; Kleinmichel, H. A new class of semismooth Newton-type methods for nonlinear complementarity problems. *Comp. Optim. Appl.* **1998**, *11*, 227–251.
17. Chen, X.; Qi, L.; Sun, D. Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. *Math. Comp.* **1998**, *67*, 519–540.
18. Kanzow, C.; Pieper, H. Jacobian smoothing methods for general nonlinear complementarity problems. *SIAM J. Optim.* **1999**, *9*, 342–372.

19. Chen, B.; Harker, P.T. Smoothing approximations to nonlinear complementarity problems. *SIAM J. Optim.* **1997**, *7*, 403–420.
20. Wu, C.Y.; Chen, G.Q. A smoothing conjugate gradient algorithm for nonlinear complementarity problems. *J. Sys. Sci. Sys. Engin.* **2008**, *17*, 460–472.
21. Clarke, F.H. *Optimization and Nonsmooth Analysis*; John Wiley and Sons, Inc.: New York, NY, USA, 1983.
22. Chen, X.J. Smoothing methods for nonsmooth, nonconvex minimization. *Math. Programm.* **2012**, *134*, 71–99.
23. Fischer, A. A special Newton-type optimization method. *Optimization* **1992**, *24*, 269–284.
24. Dai, Y.H.; Yuan, Y. An efficient hybrid conjugate gradient method for unconstrained optimization. *Ann. Oper. Res.* **2001**, *103,* 33–47.
25. Dai, Y.H. Conjugate gradient methods with Armijo-type line searches. *Acta Math. Appl. Sin.* **2002**, *18*, 123-130.
26. Xu, S. Smoothing method for minimax problem. *Comp. Optim. Appl.* **2001**, *20*, 267–279.
27. Haarala, M. Large-Scale Nonsmooth Optimization: Variable Metric Bundle Method with Limited Memory. Ph.D. Thesis, University of Jyväskylä: Jyväskylä, Finland, 13 November 2004.