GLOBAL OPTIMIZATION TECHNIQUES FOR MIXED COMPLEMENTARITY PROBLEMS

Christian Kanzow[1]

Institute of Applied Mathematics
University of Hamburg
Bundesstrasse 55
20146 Hamburg
Germany
e-mail: kanzow@math.uni-hamburg.de

July 30, 1998

Abstract. We investigate the theoretical and numerical properties of two global optimization techniques for the solution of mixed complementarity problems. More precisely, using a standard semismooth Newton-type method as a basic solver for complementarity problems, we describe how the performance of this method can be improved by combining it with a tunneling and a filled function method. These methods are tested and compared with each other on a couple of very difficult test examples.

Key words: Mixed complementarity problems, semismooth Newton method, global optimization, tunneling method, filled function method.

---

[1]Current address (October 1, 1997 — September 30, 1998): Computer Sciences Department, University of Wisconsin — Madison, 1210 West Dayton Street, Madison, WI 53706; e-mail: kanzow@cs.wisc.edu.

The research of this author was supported by the DFG (Deutsche Forschungsgemeinschaft).

1 Introduction

Given lower bounds $l_i \in \mathbb{R} \cup \{-\infty\}$ and upper bounds $u_i \in \mathbb{R} \cup \{+\infty\}$ with $l_i < u_i$ for all $i \in I := \{1, \dots, n\}$, the *mixed complementarity problem* (MCP for short) is to find a vector $x^* \in [l, u]$ such that the following implications hold for any index $i \in I$:

$$\begin{aligned} x_i^* = l_i &\implies F_i(x^*) \geq 0, \\ x_i^* \in (l_i, u_i) &\implies F_i(x^*) = 0, \\ x_i^* = u_i &\implies F_i(x^*) \leq 0; \end{aligned}$$

here, $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function and

$$l := (l_1, \dots, l_n)^T, \quad u := (u_1, \dots, u_n)^T.$$

Since it is easy to see that x^* is a solution of MCP if and only if it satisfies the variational inequality

$$F(x^*)^T(x - x^*) \geq 0 \quad \text{for all } x \in [l, u],$$

the mixed complementarity problem is sometimes also called the *box constrained variational inequality problem*. Note that it reduces to the standard nonlinear complementarity problem of finding a feasible point for the system

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

if $l_i = 0$ and $u_i = +\infty$ for all $i \in I$, and that we obtain the problem of finding a solution of the nonlinear system of equations

$$F(x) = 0$$

if $l_i = -\infty$ and $u_i = +\infty$ for all $i \in I$.

Apart from these special cases, the mixed complementarity problem has a whole bunch of other applications ranging from operations research to economic equilibrium and engineering problems, and we refer the interested reader to the recent article [12] by Ferris and Pang for an overview. Due to the importance of the mixed complementarity problem, many researchers try to find efficient and robust methods for the solution of complementarity problems. In fact, we believe that much progress has been made in this area during the last decade so that we are able to solve much more complicated problems now than just a few years ago.

So far, almost all of these methods try to minimize a certain *merit function* for the MCP which, typically, is zero at a solution and positive otherwise. Hence, minimizing such a merit function is a global optimization problem. In fact, there are two steps where global optimization plays a significant role.

The first step is the choice of a good merit function. This is highly important because the right choice of a merit function may already avoid many problems with local-nonglobal minimizers since one merit function may have local-nonglobal minimizers which another merit function might not have. For example, it is known that the implicit Lagrangian [21] is a merit function of the standard nonlinear complementarity problem which might have local-nonglobal minimizers even for strictly monotone problems [29]. On the other hand, it is known that all local minimizers are already global ones for, e.g., the Fischer-Burmeister merit function [13, 16, 9] if the complementarity problem is strictly monotone or just monotone. However, one cannot expect to construct a merit function such that all minimizers are global without requiring any monotonicity-type conditions on F.

We therefore believe that it is no longer possible to get substantial improvements by introducing new merit functions. Instead, our feeling is that one has to consider algorithms which are able to deal with the problem that even the currently best merit functions might have local-nonglobal minimizers. And this is the second time where global optimization plays a central role.

There are actually a couple of papers available which use ideas from global optimization in order to solve complementarity and related problems. Most of these papers deal with homotopy methods, see, e.g., Watson [28], Sellami and Robinson [24, 25] and Billups [3]. While these methods seem to be fairly robust, they usually need an enourmous number of iterations in order to trace the underlying homotopy path. On the other hand, Kostreva and Zheng [18] describe an integral optimization method for the solution of the standard nonlinear complementarity problem, but they mainly focus on problems where the function F is not necessarily differentiable (not even continuous).

In this paper, we discuss the theoretical and numerical properties of two different global optimization techniques for the solution of mixed complementarity problems. The two techniques investigated here are the tunneling approach [20] and the filled function method [15]. We describe the influence of these techniques in combination with one particular class of methods for the solution of mixed complementarity problems, namely the class of semismooth solvers for MCP. As known to the author, none of these techniques have been used earlier in the complementarity community.

The organization of this paper is as follows: In Section 2 we first give a review of a specific semismooth Newton-type method for the solution of the mixed complementarity problem. This method is used as the basic solver in this paper. We stress, however, that many other methods for MCP can be used as the basic solver. We then describe a switching criterion in Section 3, i.e., a criterion which indicates when this semismooth solver is likely to fail on the given example so that it seems advantageous to switch to a global optimization method. Sections 4 and 5 then describe the main ideas of the tunneling and the filled function methods. In fact, we also describe an extension of the original filled function idea by Ge [15]. We then present extensive numerical results in Section 6 and conclude this paper with some final remarks in Section 7.

The notation used in this paper is rather standard: The n-dimensional Euclidean space is denoted by $\mathbb{R}^n$, with scalar product $x^T y$ for two vectors $x, y \in \mathbb{R}^n$. The symbol $\|\cdot\|$ denotes the Euclidean vector norm or its associated matrix norm, whereas $\|x\|_\infty$ is the maximum norm in $\mathbb{R}^n$. Finally, x_+ denotes the projection of a vector $x \in \mathbb{R}^n$ onto the box $[l, u]$.

2 Basic Semismooth Newton-type Method

In this section we recall the basic facts about one of the standard methods for the solution of the mixed complementarity problem. This method belongs to the class of semismooth Newton-type methods and is, basically, a nonsmooth Newton method applied to a reformulation of the MCP as a nonlinear (and nonsmooth) system of equations.

In the following, we give a short description of this reformulation, see [1, 10] for further details. To this end, let us define the Fischer-Burmeister function $\phi : \mathbb{R}^2 \to \mathbb{R}$ by

$$\phi(a, b) := \sqrt{a^2 + b^2} - a - b,$$

see [13], and let us introduce a partition of the index set I:

$$\begin{aligned} I_l &:= \{i \in I \mid -\infty < l_i < u_i = +\infty\}, \\ I_u &:= \{i \in I \mid -\infty = l_i < u_i < +\infty\}, \\ I_{lu} &:= \{i \in I \mid -\infty < l_i < u_i < +\infty\}, \\ I_f &:= \{i \in I \mid -\infty = l_i < u_i = +\infty\}, \end{aligned}$$

i.e., I_l, I_u, I_{lu} and I_f denote the set of indices $i \in I$ with finite lower bounds only, finite upper bounds only, finite lower and upper bounds and no finite bounds on the variable x_i, respectively. Hence the subscripts in the above index sets indicate which bounds are finite, with the only exception of I_f which contains the free variables.

We now define an operator $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ componentwise as follows:

$$\Phi_i(x) := \begin{cases} \phi(x_i - l_i, F_i(x)) & \text{if } i \in I_l, \\ -\phi(u_i - x_i, -F_i(x)) & \text{if } i \in I_u, \\ \phi(x_i - l_i, \phi(u_i - x_i, -F_i(x))) & \text{if } i \in I_{lu}, \\ -F_i(x) & \text{if } i \in I_f. \end{cases}$$

Then it is easy to see that

$$x^* \text{ solves MCP} \iff x^* \text{ solves } \Phi(x) = 0.$$

A similar reformulation of the mixed complementarity problem, also based on the Fischer-Burmeister function, is described in the paper [27] by Sun and Womersley. Further reformulations can be obtained by replacing the Fischer-Burmeister function by, e.g., the recently introduced function from [6] which seems to have somewhat stronger theoretical properties and a better numerical behaviour. For the sake of simplicity, however, we will only consider the reformulation introduced at the beginning of this section for the theoretical part of this paper. The interested reader can easily extend our results to other appropriate reformulations.

In order to solve the mixed complementarity problem, we now apply a nonsmooth Newton method from Qi [22] to the system $\Phi(x) = 0$ and globalize it using the corresponding merit function

$$\Psi(x) := \frac{1}{2}\Phi(x)^T\Phi(x).$$

Note that this function is continuously differentiable everywhere [1, 10] with

$$\nabla\Psi(x) = H^T\Phi(x), \tag{1}$$

where $H \in \mathbb{R}^{n\times n}$ denotes an arbitrary element from the B-subdifferential $\partial_B\Phi(x)$, see [10, 22] for a definition of this set.

Algorithm 2.1 *(Semismooth Newton-type Method)*

(S.0) (Initialization)
Choose $x^0 \in \mathbb{R}^n, \rho > 0, \beta \in (0,1), \sigma \in (0,1/2), p > 2,$ *and set* $k := 0$.

(S.1) (Stopping Criterion)
If x^k *satisfies a suitable termination criterion: STOP.*

(S.2) (Search Direction Calculation)
Select an element $H_k \in \partial_B \Phi(x^k)$. *Find a solution* $d^k \in \mathbb{R}^n$ *of the linear system*

$$H_k d = -\Phi(x^k). \tag{2}$$

If this system is not solvable or if the descent condition

$$\nabla \Psi(x^k)^T d^k \leq -\rho \|d^k\|^p \tag{3}$$

is not satisfied, set $d^k := -\nabla \Psi(x^k)$.

(S.3) (Line Search)
Compute $t_k := \max\{\beta^\ell \mid \ell = 0, 1, 2, \ldots\}$ *such that*

$$\Psi(x^k + t_k d^k) \leq \Psi(x^k) + t_k \nabla \Psi(x^k)^T d^k.$$

(S.4) (Update)
Set $x^{k+1} := x^k + t_k d^k, k \leftarrow k+1,$ *and go to (S.1).*

We next discuss the convergence properties of Algorithm 2.1. These properties can be derived from the two papers [7] by De Luca, Facchinei and Kanzow and [10] by Ferris, Kanzow and Munson, and we therefore refer the interested reader to these papers for some further details. Our first result deals with the global convergence properties of Algorithm 2.1.

Theorem 2.2 *Every accumulation point of a sequence* $\{x^k\}$ *generated by Algorithm 2.1 is a stationary point of* Ψ, *and such a stationary point is a solution of MCP under relatively mild assumptions.*

Before summarizing the local convergence properties of Algorithm 2.1 in our next result, we recall that a solution x^* of MCP is said to be a BD-regular solution of the system $\Phi(x) = 0$ if all elements $H \in \partial_B \Phi(x^*)$ are nonsingular. For example, if x^* is a strongly regular solution of MCP in the sense of Robinson [23], then x^* is a BD-regular solution of $\Phi(x) = 0$, see [10].

Theorem 2.3 *Let* $x^* \in \mathbb{R}^n$ *be a BD-regular solution of* $\Phi(x) = 0$ *and assume that* x^* *is an accumulation point of a sequence* $\{x^k\}$ *generated by Algorithm 2.1. Then the following statements hold:*

(1) The entire sequence $\{x^k\}$ *converges to* x^*.

(2) The search direction d^k *is eventually given by the Newton equation (2).*

(3) The full stepsize $t_k = 1$ is eventually accepted in Step (S.3).

(4) The rate of convergence is Q-superlinear.

(5) If F' is locally Lipschitzian, then the rate of convergence is Q-quadratic.

For later reference, we will make statements (2) and (3) of Theorem 2.3 more precise in the following remark.

Remark 2.4 *Let $x^* \in \mathbb{R}^n$ be a BD-regular solution of $\Phi(x) = 0$. Then there is a neighbourhood of x^* such that, whenever x^k belongs to this neighbourhood, the linear system (2) has a unique solution d^k which satisfies the descent condition (3). Moreover, the full stepsize $t_k = 1$ will be accepted for this descent direction by the Armijo-rule in Step (S.3).*

3 Switching Criterion

In this section we investigate the following question: When shall we terminate Algorithm 2.1 if we run into troubles? This question is highly important from a practical point of view because if we are able to identify (hopefully relatively early, but not too early) that Algorithm 2.1 may not be able to solve a certain test example, we may want to switch to another method (e.g., a global optimization method) hoping that this method will make more progress.

Our answer to this question is given in the following algorithm. It is a modified semismooth Newton-type method which differs from Algorithm 2.1 mainly in the introduction of a new termination criterion in Step (S.3). The idea of this new termination check is that, if one of the two conditions given there is satisfied, then something is going wrong, so we stop the iteration. Later we will discuss the problem of what can be done if Step (S.3) becomes active.

Note that the new termination check in Step (S.3) is related to the one introduced by Billups and Ferris in their paper [4], where it is used in order to improve the global convergence properties of a QP-based method for the solution of complementarity problems.

Algorithm 3.1 *(Modified Semismooth Newton-type Method)*

(S.0) (Initialization)
Choose $x^0 \in \mathbb{R}^n, \rho > 0, \beta \in (0,1), \sigma \in (0,1/2), p > 2, \delta \in (0,2), \Delta > 0,$ and set $k := 0$.

(S.1) (Termination Criterion)
If x^k is a solution of MCP: STOP.

(S.2) (Search Direction Calculation)
Select an element $H_k \in \partial_B \Phi(x^k)$. Find a solution $d^k \in \mathbb{R}^n$ of the linear system

$$H_k d = -\Phi(x^k). \tag{4}$$

If this system is not solvable or if the descent condition

$$\nabla \Psi(x^k)^T d^k \leq -\rho \|d^k\|^p \tag{5}$$

is not satisfied, set $d^k := -\nabla \Psi(x^k)$.

(S.3) (Check for Failure)
If $\nabla\Psi(x^k)^T d^k \geq -\delta\Psi(x^k)$ or if $\|d^k\| \geq \Delta$, then terminate the algorithm, returning the point x^k along with a failure message; otherwise, continue.

(S.4) (Line Search)
Compute $t_k := \max\{\beta^\ell \mid \ell = 0, 1, 2, \ldots\}$ such that

$$\Psi(x^k + t_k d^k) \leq \Psi(x^k) + t_k \sigma \nabla\Psi(x^k)^T d^k.$$

(S.5) (Update)
Set $x^{k+1} := x^k + t_k d^k, k \leftarrow k+1$, and go to (S.1).

In our subsequent analysis, we always assume implicitly that Algorithm 3.1 does not terminate after finitely many iterations in Step (S.1).

We now want to show that Algorithm 3.1 has the same local convergence properties as Algorithm 2.1. To this end, we prove that none of the tests in Step (S.3) of Algorithm 3.1 is satisfied if we are sufficiently close to a BD-regular solution of the system $\Phi(x) = 0$. We first show that the first test in Step (S.3) is never active as long as d^k is the Newton direction calculated as the solution of the linear system (4).

Lemma 3.2 *If x^k is not a solution of MCP and $d^k \in \mathbb{R}^n$ is computed from (4), then $\nabla\Psi(x^k)^T d^k < -\delta\Psi(x^k)$.*

Proof. From (1), we have $\nabla\Psi(x^k) = H_k^T \Phi(x^k)$, where H_k denotes the matrix from the linear system (4). On the other hand, we have $H_k d^k = -\Phi(x^k)$ from (4). Since $\Psi(x^k) = \frac{1}{2}\Phi(x^k)^T\Phi(x^k)$, we therefore get

$$\nabla\Psi(x^k)^T d^k = \Phi(x^k)^T H_k d^k = -\Phi(x^k)^T\Phi(x^k) = -2\Psi(x^k) < -\delta\Psi(x^k),$$

where the last inequality follows from $\delta \in (0, 2)$ and $\Psi(x^k) > 0$. □

Note that Lemma 3.2 does not even assume that the matrix H_k in the Newton equation (4) is nonsingular; instead, it just assumes that the linear system (4) is consistent.

The next result shows that the first test in Step (S.3) of Algorithm 3.1 is never satisfied if we are sufficiently close to a BD-regular solution of $\Phi(x) = 0$.

Lemma 3.3 *Let $x^* \in \mathbb{R}^n$ be a BD-regular solution of $\Phi(x) = 0$. Then there is a neighbourhood of x^* such that*

$$\nabla\Psi(x^k)^T d^k < -\delta\Psi(x^k)$$

for all x^k in this neighbourhood, where d^k denotes the search direction computed by Algorithm 3.1.

Proof. It follows from Remark 2.4 that, if x^k is sufficiently close to x^*, then $d^k \in \mathbb{R}^n$ can be computed from (4) and satisfies the sufficient decrease condition (5). Hence, in a sufficiently small neighbourhood of x^*, d^k is always the Newton direction, so the assertion follows from Lemma 3.2. □

We next show that also the second test in Step (S.3) of Algorithm 3.1 is never met in a small neighbourhood of a BD-regular solution of $\Phi(x) = 0$.

Lemma 3.4 *Let $x^* \in \mathbb{R}^n$ be a BD-regular solution of $\Phi(x) = 0$. Then there is a neighbourhood of x^* such that*

$$\|d^k\| < \Delta$$

for all x^k in this neighbourhood, where d^k denotes the search direction computed by Algorithm 3.1.

Proof. Since x^* is a BD-regular solution of the system $\Phi(x) = 0$, it follows from Lemma 2.6 in Qi [22] that there is a constant $c > 0$ with $\|H_k^{-1}\| \le c$ for all $H_k \in \partial_B \Phi(x^k)$ and all x^k in a sufficiently small neighbourhood of x^*. We therefore obtain from (4) that

$$\|d^k\| \le \|H_k^{-1}\| \, \|\Phi(x^k)\| \le c\|\Phi(x^k)\| \to 0$$

for $x^k \to x^*$. This immediately implies the statement of our lemma. $\square$

We obtain as a consequence of the previous results that Algorithm 3.1 eventually coincides with Algorithm 2.1 if we are close enough to a BD-regular solution of $\Phi(x) = 0$. Hence all statements on the local rate of convergence as given in Theorem 2.3 also hold for Algorithm 3.1.

We now turn to the global convergence properties of Algorithm 3.1. The central part for our main global convergence result is contained in the following statement.

Proposition 3.5 *If Algorithm 3.1 does not terminate after finitely many iterations, then $\{\Psi(x^k)\} \to 0$ (i.e., $\{x^k\}$ is a minimizing sequence for Ψ) or the entire sequence $\{x^k\}$ converges to a unique point x^* (which is not necessarily a solution of MCP).*

Proof. In view of our assumption, neither the test in Step (S.1) nor the tests in Step (S.3) of Algorithm 3.1 are satisfied for a finite index k. In particular, we have $\nabla\Psi(x^k)^T d^k < -\delta\Psi(x^k)$ for all k. Hence, it follows from our line search rule in Step (S.4) that

$$\begin{aligned} \Psi(x^{k+1}) &\le \Psi(x^k) + t_k \sigma \nabla\Psi(x^k)^T d^k \\ &< \Psi(x^k) - \sigma\delta t_k \Psi(x^k) \end{aligned}$$

and therefore

$$\alpha_k \Psi(x^k) < \Psi(x^k) - \Psi(x^{k+1}), \tag{6}$$

where $\alpha_k := \sigma\delta t_k > 0$.

Since the sequence $\{\Psi(x^k)\}$ is obviously monotonically decreasing and bounded from below, it converges to a scalar $\Psi^* \ge 0$. If $\Psi^* = 0$, then $\{\Psi(x^k)\} \to 0$ as stated. So consider the case where $\Psi^* > 0$. It follows from (6) that

$$\sum_{j=0}^{k} \alpha_j \Psi(x^j) < \Psi(x^0) - \Psi(x^{k+1}).$$

Since $\{\Psi(x^k)\} \to \Psi^*$, this implies

$$\sum_{k=0}^{\infty} \alpha_k \Psi(x^k) \le \Psi(x^0) - \Psi^* < \infty.$$

On the other hand, we have $\Psi(x^k) \geq \Psi^*$ for all k, so that

$$\sum_{k=0}^{\infty} \alpha_k \Psi^* \leq \sum_{k=0}^{\infty} \alpha_k \Psi(x^k) < \infty$$

and therefore

$$\sum_{k=0}^{\infty} \alpha_k < \infty$$

since $\Psi^* > 0$. In view of the definition of α_k, this means that the series $\sum_{k=0}^{\infty} t_k$ is also finite. Since our algorithm does not terminate after finitely many iterations, we also have $\|d^k\| < \Delta$ for all k, cf. the second test in Step (S.3). We therefore obtain

$$\sum_{k=0}^{\infty} t_k \|d^k\| < \infty. \tag{7}$$

Since $x^{k+1} = x^k + t_k d^k$ for all k, this implies that $\{x^k\}$ is a Cauchy sequence. Hence $\{x^k\}$ is convergent. $\square$

We are now in the position to state and prove the main global convergence result for Algorithm 3.1 which says that this method either terminates after finitely many iterations in Step (S.3) or that it generates a minimizing sequence for our merit function Ψ so that, in particular, any accumulation point is a solution of the MCP.

Theorem 3.6 *Either Algorithm 3.1 generates a sequence $\{x^k\}$ such that $\{\Psi(x^k)\} \to 0$ or it terminates after finitely many iterations in Step (S.3).*

Proof. We first recall that, in view of our general assumption, Algorithm 3.1 does not terminate in Step (S.1) after finitely many iterations. Assume now that Algorithm 3.1 generates an infinite sequence $\{x^k\}$, i.e., also the termination criteria in Step (S.3) are not active at any iteration k. Using Proposition 3.5, we then have $\{\Psi(x^k)\} \to 0$ or $\{x^k\}$ converges to a unique point x^*. In the first case we are done.

In the second case, the vector x^* is a stationary point of Ψ; this follows from the observation that x^* is a limit point of the sequence $\{x^k\}$ and that Algorithm 3.1 is identical to Algorithm 2.1 (recall, by our assumption, that Step (S.3) in Algorithm 3.1 is never active), so that we can apply Theorem 2.2 in this situation.

Now, by continuity, we have $\Psi(x^k) \to \Psi(x^*)$. If $\Psi(x^*) = 0$, then $\{x^k\}$ is a minimizing sequence of Ψ and there is nothing to prove any more. Otherwise we have $\Psi(x^*) > 0$. Since, however, $\nabla\Psi(x^*) = 0$ and the sequence $\{\|d^k\|\}$ is bounded (otherwise the second test in Step (S.3) would be active), we have

$$\nabla\Psi(x^k)^T d^k \geq -\delta\Psi(x^k)$$

for all large k, i.e., the first test in Step (S.3) is satisfied.

Hence we see that, in either case, our algorithm either generates a minimizing sequence for Ψ or terminates after finitely many iterations in Step (S.3). $\square$

It is interesting to note that Theorem 3.6 holds without any assumptions on the MCP. Since everything is fine if Algorithm 3.1 generates a minimizing sequence for Ψ, the critical case is when this method terminates in Step (S.3). If this happens, we have to generate in some way a better point. We state this formally in the next algorithm which differs from Algorithm 3.1 mainly in Step (S.3).

Algorithm 3.7 *(Global Semismooth Newton-type Method)*

(S.0) *(Initialization)*
Choose $x^0 \in \mathbb{R}^n, \rho > 0, \beta \in (0,1), \sigma \in (0,1/2), p > 2, \delta \in (0,2), \Delta > 0, \gamma \in (0,1)$ *and set* $k := 0$.

(S.1) *(Termination Criterion)*
If x^k *is a solution of MCP: STOP.*

(S.2) *(Search Direction Calculation)*
Select an element $H_k \in \partial_B \Phi(x^k)$. *Find a solution* $d^k \in \mathbb{R}^n$ *of the linear system*

$$H_k d = -\Phi(x^k).$$

If this system is not solvable or if the descent condition

$$\nabla \Psi(x^k)^T d^k \leq -\rho \|d^k\|^p$$

is not satisfied, set $d^k := -\nabla \Psi(x^k)$.

(S.3) *(Improve Current Point)*
If $\nabla \Psi(x^k)^T d^k \geq -\delta \Psi(x^k)$ *or if* $\|d^k\| \geq \Delta$, *then generate a point* x^{k+1} *with* $\Psi(x^{k+1}) \leq \gamma \Psi(x^k)$, *set* $k \leftarrow k+1$ *and go to (S.1).*

(S.4) *(Line Search)*
Compute $t_k := \max\{\beta^\ell \mid \ell = 0, 1, 2, \ldots\}$ *such that*

$$\Psi(x^k + t_k d^k) \leq \Psi(x^k) + t_k \sigma \nabla \Psi(x^k)^T d^k.$$

(S.5) *(Update)*
Set $x^{k+1} := x^k + t_k d^k, k \leftarrow k+1$, *and go to (S.1).*

It follows immediately from Theorem 3.6 that any sequence $\{x^k\}$ generated by Algorithm 3.7 is a minimizing sequence for the merit function Ψ. However, so far it is not clear how to generate the improved point x^{k+1} in Step (S.3). Two possibilities of how this can be accomplished will be discussed in more detail in the next two sections.

4 Tunneling Methods

Throughout this section, let x^* be a local-nonglobal minimizer of Ψ or at least any vector at which our basic semismooth solver from Algorithm 2.1 seems to run into troubles (e.g., in the sense that one of our tests in Step (S.3) in Algorithm 3.7 is satisfied). The main idea of a tunneling method is to introduce a new function which has a pole at x^* and which can therefore be minimized in order to escape from this critical point. This idea was first used by Levy and Montalvo [20] (for optimization problems) and later extended to (smooth) nonlinear systems of equations by Levy and Gómez [19].

Here we introduce two different tunneling functions. The first one is the classical tunneling function from [20, 19]. It is defined by

$$T_\Psi^c(x) := \frac{1}{2} T_\Phi^c(x)^T T_\Phi^c(x) = \frac{1}{2}\|T_\Phi^c(x)\|^2,$$

where the operator $T_\Phi : \mathbb{R}^n \setminus \{x^*\} \to \mathbb{R}^n$ is given by

$$T_\Phi^c(x) := \frac{\Phi(x)}{\|x - x^*\|}.$$

Note that a vector is a solution of the MCP if and only if it is a solution of $T_\Phi^c(x) = 0$. Moreover, the classical tunneling function can be rewritten as

$$T_\Psi^c(x) = \frac{\Psi(x)}{\|x - x^*\|^2},$$

and this shows that $T_\Psi^c(x)$ itself is continuously differentiable on $\mathbb{R}^n \setminus \{x^*\}$ since Ψ is smooth everywhere. Hence we can apply our basic semismooth solver from Algorithm 2.1 to the nonsmooth system of equations

$$T_\Phi^c(x) = 0$$

in order to minimize the classical tunneling function.

The second tunneling function to be considered in this paper is the exponential tunneling function introduced in [17]. It is defined by

$$T_\Psi^e(x) := \frac{1}{2} T_\Phi^e(x)^T T_\Phi^e(x) = \frac{1}{2}\|T_\Phi^e(x)\|^2,$$

where

$$T_\Phi^e(x) := \Phi(x) \exp\left(\frac{1}{\|x - x^*\|^2}\right).$$

Again, we see that solving the MCP is equivalent to solving the nonlinear system of equations $T_\Phi^e(x) = 0$; furthermore, since

$$T_\Psi^e(x) = \Psi(x) \left[\exp\left(\frac{2}{\|x - x^*\|^2}\right)\right],$$

we see that also the exponential tunneling function is continuously differentiable on $\mathbb{R}^n \setminus \{x^*\}$. Hence, also the exponential tunneling function can be minimized by applying the basic semismooth solver from Algorithm 2.1 to the nonlinear system of equations

$$T_\Phi^e(x) = 0.$$

Both the classical and the exponential tunneling functions have their drawbacks, however, they provide some reasonable (though heuristic) ways to escape from a local-nonglobal minimum of the underlying merit function Ψ. Moreover, minimizing one of these tunneling functions corresponds to solving the original MCP.

5 Filled Function Methods

The basic idea of using a filled function method is similar to the one of a tunneling method: If x^* denotes a local-nonglobal minimum of our merit function Ψ, it tries to escape from this minimum by using a new function. In contrast to tunneling methods, the filled function methods do not place an infinite pole at the point x^* but introduce a new function which, instead of having a minimum at x^*, has a maximum at this point and can therefore be further minimized. This idea goes back to Ge [15].

In the following, we slightly extend the idea by Ge [15]. To this end, let us introduce a one-dimensional function

$$\theta : \mathbb{R} \to \mathbb{R}$$

having the following properties:

(P.1) $\theta(t) > 0$ for all $t \in \mathbb{R}$;

(P.2) $\theta(0) = 1$;

(P.3) $\theta(t) \leq 1$ for all $t \in \mathbb{R}$.

Basically, these conditions say that θ is any positive and bounded function which takes its maximum value in the origin. We next give two simple examples.

Example 5.1 *The following functions $\theta : \mathbb{R} \to \mathbb{R}$ satisfy properties (P.1)–(P.3):*

(a) $\theta(t) := exp(-t^2)$*;*

(b) $\theta(t) := 1/(1 + t^2)$.

Now let $r > 0, \rho > 0$ be any given constants and θ be any function satisfying properties (P.1)–(P.3). Then define

$$P_\theta(x; r, \rho) := \frac{1}{\Psi(x) + r} \theta(\|x - x^*\|/\rho), \tag{8}$$

where x^* denotes the local-nonglobal minimum of our merit function Ψ. The function $P_\theta(\cdot; r, \rho)$ is called a filled function. If θ is the function from Example 5.1 (a), we obtain

$$P_1(x) := P_\theta(x; r, \rho) = \frac{1}{\Psi(x) + r} \exp(-\|x - x^*\|^2/\rho^2),$$

which is precisely the function considered by Ge [15], whereas we get

$$P_2(x) := P_\theta(x; r, \rho) = \frac{1}{\Psi(x) + r} \cdot \frac{1}{1 + \|x - x^*\|^2/\rho^2}$$

when using the function θ from Example 5.1 (b). Generalizing Theorem 2.1 in Ge [15], we can easily prove the following result which explains to some extent the name filled function for $P_\theta(\cdot; r, \rho)$.

Proposition 5.2 *Let θ be any function satisfying properties (P.1)–(P.3) and define P_θ by (8). Then the following statements hold:*

(a) If x^* is a local minimizer of Ψ, then x^* is a local maximizer of P_θ.

(b) If x^* is a strict local minimizer of Ψ, then x^* is a strict local maximizer of P_θ.

Proof. We only prove part (a) since the proof of part (b) is very similar.

So assume that x^* is a local minimizer of Ψ. Then

$$\Psi(x) \geq \Psi(x^*)$$

for all $x \in \mathbb{R}^n$ sufficiently close to x^*. Hence we obtain for all these x by using properties (P.1)–(P.3):

$$\begin{aligned} P_\theta(x^*; r, \rho) &= \frac{1}{\Psi(x^*) + r} \\ &\geq \frac{1}{\Psi(x) + r} \\ &\geq \frac{1}{\Psi(x) + r} \theta(\|x - x^*\|/\rho) \\ &= P_\theta(x; r, \rho). \end{aligned}$$

This shows that x^* is a local maximizer of $P_\theta(\cdot; r, \rho)$. □

In contrast to the tunneling methods, the filled function methods have the advantage that we do not have to deal with any numerical difficulties which may arise around the possible pole x^* of the tunneling functions and that the filled function methods provide a completely nonheuristic way to escape from local-nonglobal minima. On the other hand, the filled function methods have severe disadvantages: Minimizing a filled function does not guarantee that we find a solution of the underlying complementarity problem, and, furthermore, the computational overhead is more significant: We cannot apply our basic semismooth solver in order to minimize a filled function (since there is no corresponding equation formulation), so we have to implement an unconstrained minimization method in order to minimize a filled function P_θ. Since P_θ is only once but not twice continuously differentiable, we can only use a first-order method. Second-order methods are also prohibited since, in general, the second derivatives of the mapping F are not available.

6 Numerical Results

In this section, we present some numerical results for the different global optimization techniques discussed in this paper. To this end, we implemented our basic semismooth solver from Algorithm 2.1 in MATLAB using the parameters

$$\rho = 10^{-10}, \quad \beta = 0.5, \quad \sigma = 10^{-4}, \quad \text{and} \quad p = 2.1.$$

The termination criterion is

$$\|r(x)\|_\infty \le 10^{-6},$$

where $r(x) := x - [x - F(x)]_+$ denotes the natural residual of the MCP, but we also stop the algorithm if the number of iterations exceeds 500. We use the constants

$$\delta = 10^{-8}, \quad \Delta = n \cdot 10^8, \quad \text{and} \quad \gamma = 0.9$$

in Step (S.3) of Algorithm 3.7; in addition, we switch to the global optimization technique if the stepsize t_k becomes too small.

Finally, we use the Polak-Ribière conjugate gradient method with restarts and a line search satisfying the strong Wolfe-Powell conditions in order to minimize the two filled functions from the previous section, see [14] for further details.

All our test problems are taken from the GAMSLIB and the MCPLIB, see [5, 8]. Instead of using all examples from these two test problem collections, we decided to present numerical results only for those problems which are usually being regarded as very difficult. Furthermore, all test examples selected here are of dimension $n \le 150$. Obviously, this is a restriction, but it is our impression that neither the tunneling methods nor the filled function methods are very reliable for larger problems.

We run our program on a SUN SPARC station and summarize the numerical results in Tables 1 – 3, with Table 1 containing the results for the basic semismooth solver from Algorithm 2.1, Table 2 containing the results for the modification using the two different tunneling approaches, and Table 3 containing the results for the two different filled function modifications (here, we call the filled functions P_1 and P_2 the exponential and the rational filled function, respectively). The columns of these tables have the following meanings:

problem:	name of the test problem in GAMSLIB/MCPLIB
n:	dimension of the test problem
SP:	number of starting point used
k_{semi}:	number of iterations used in the basic semismooth solver
k_{total}:	total number of iterations used
j_{global}:	number of times we switch to the global optimization technique
$\|r(x^f)\|_\infty$:	norm of the natural residual at the final iterate x^f.

Note that the difference between k_{total} and k_{semi} gives the number of iterations used in the global optimization techniques. More precisely, if the global optimization phase becomes active more than once (i.e., if $j_{global} > 1$), this difference provides the cumulated number of these iterations.

We next discuss the results given in Tables 1 – 3: The results in Table 1 are basically there in order to compare our global optimization techniques with the basic solver. The only

Table 1: Numerical results for the basic semismooth solver

problem	n	SP	k_{semi}	$\|r(x^f)\|_\infty$
billups	1	1	—	—
billups	1	2	—	—
billups	1	3	—	—
colvdual	20	1	14	1.1e-7
colvdual	20	2	37	7.5e-10
colvdual	20	3	10	1.7e-11
colvdual	20	4	—	—
ehl_k40	41	1	36	5.3e-8
ehl_k60	61	1	38	2.8e-8
ehl_k80	81	1	45	9.2e-9
ehl_kost	101	1	27	6.6e-7
ehl_kost	101	2	26	1.2e-9
ehl_kost	101	3	26	1.2e-9
pgvon105	105	1	48	5.0e-10
pgvon105	105	2	—	—
pgvon105	105	3	—	—
pgvon106	106	1	—	—
pgvon106	106	2	—	—
pgvon106	106	3	—	—
scarfbnum	39	1	21	3.3e-8
scarfbnum	39	2	20	2.8e-9
scarfbsum	40	1	20	3.6e-10
scarfbsum	40	2	19	3.3e-10
vonthmcp	125	1	48	1.6e-7
vonthmge	80	1	—	—
duopoly	63	1	—	—
simple-ex	17	1	—	—
games	16	1	13	3.3e-8
games	16	2	14	5.6e-8
games	16	3	—	—
games	16	4	20	1.1e-10
games	16	5	—	—
games	16	6	22	2.4e-7
games	16	7	24	9.0e-9
games	16	8	17	1.2e-10
games	16	9	23	2.0e-9
games	16	10	18	1.5e-7

Table 2: Numerical results for the tunneling methods

		classical tunneling				exponential tunneling			
problem	SP	k_{semi}	k_{total}	j_{global}	$\|r(x^f)\|_\infty$	k_{semi}	k_{total}	j_{global}	$\|r(x^f)\|_\infty$
billups	1	9	12	1	4.7e-11	9	13	1	9.9e-12
billups	2	13	16	1	4.7e-11	13	17	1	9.9e-12
billups	3	11	14	1	4.7e-11	11	15	1	9.9e-12
colvdual	1	14	14	0	1.1e-7	14	14	0	1.1e-7
colvdual	2	37	37	0	7.5e-10	37	37	0	7.5e-10
colvdual	3	10	10	0	1.7e-11	10	10	0	1.7e-11
colvdual	4	295	407	4	2.6e-9	77	89	1	2.4e-10
ehl_k40	1	36	36	0	5.3e-8	36	36	0	5.3e-8
ehl_k60	1	38	38	0	2.8e-8	38	38	0	2.8e-8
ehl_k80	1	45	45	0	9.2e-9	45	45	0	9.2e-9
ehl_kost	1	27	27	0	6.6e-7	27	27	0	6.6e-7
ehl_kost	2	26	26	0	1.2e-9	26	26	0	1.2e-9
ehl_kost	3	26	26	0	1.2e-9	26	26	0	1.2e-9
pgvon105	1	48	48	0	5.0e-10	48	48	0	5.0e-10
pgvon105	2	182	186	1	5.6e-7	185	189	1	9.3e-7
pgvon105	3	155	159	1	8.7e-7	157	161	1	5.6e-7
pgvon106	1	99	244	2	7.3e-7	—	—	—	—
pgvon106	2	201	260	1	8.8e-7	—	—	—	—
pgvon106	3	345	478	1	8.4e-7	—	—	—	—
scarfbnum	1	21	21	0	3.3e-8	21	21	0	3.3e-8
scarfbnum	2	20	20	0	2.8e-9	20	20	0	2.8e-9
scarfbsum	1	20	20	0	3.6e-10	20	20	0	3.6e-10
scarfbsum	2	19	19	0	3.3e-10	19	19	0	3.3e-10
vonthmcp	1	35	39	1	1.1e-8	35	40	1	2.5e-8
vonthmge	1	27	35	1	1.8e-12	35	37	1	6.1e-7
duopoly	1	—	—	—	—	144	229	3	5.2e-8
simple-ex	1	25	34	1	6.9e-7	20	24	1	5.5e-7
games	1	13	13	0	3.3e-8	13	13	0	3.3e-8
games	2	14	14	0	5.6e-8	14	14	0	5.6e-8
games	3	—	—	—	—	113	144	3	9.6e-7
games	4	20	20	0	1.1e-10	20	20	0	1.1e-10
games	5	—	—	—	—	123	141	1	8.4e-7
games	6	22	22	0	2.4e-7	22	22	0	2.4e-7
games	7	24	24	0	9.0e-9	24	24	0	9.0e-9
games	8	17	17	0	1.2e-10	17	17	0	1.2e-10
games	9	23	23	0	2.0e-9	23	23	0	2.0e-9
games	10	18	18	0	1.5e-7	18	18	0	1.5e-7

Table 3: Numerical results for the filled function methods

		exponential filled function				rational filled function			
problem	SP	k_{semi}	k_{total}	j_{global}	$\|r(x^f)\|_\infty$	k_{semi}	k_{total}	j_{global}	$\|r(x^f)\|_\infty$
billups	1	11	12	1	4.1e-12	14	15	1	5.7e-8
billups	2	16	17	1	6.1e-8	20	21	1	7.9e-8
billups	3	14	15	1	5.8e-8	18	19	1	7.0e-8
colvdual	1	14	14	0	1.1e-7	14	14	0	1.1e-7
colvdual	2	37	37	0	7.5e-10	37	37	0	7.5e-10
colvdual	3	10	10	0	1.7e-11	10	10	0	1.7e-11
colvdual	4	—	—	—	—	—	—	—	—
ehl_k40	1	36	36	0	5.3e-8	36	36	0	5.3e-8
ehl_k60	1	38	38	0	2.8e-8	38	38	0	2.8e-8
ehl_k80	1	45	45	0	9.2e-9	45	45	0	9.2e-9
ehl_kost	1	27	27	0	6.6e-7	27	27	0	6.6e-7
ehl_kost	2	26	26	0	1.2e-9	26	26	0	1.2e-9
ehl_kost	3	26	26	0	1.2e-9	26	26	0	1.2e-9
pgvon105	1	48	48	0	5.0e-10	48	48	0	5.0e-10
pgvon105	2	—	—	—	—	—	—	—	—
pgvon105	3	—	—	—	—	—	—	—	—
pgvon106	1	—	—	—	—	—	—	—	—
pgvon106	2	—	—	—	—	—	—	—	—
pgvon106	3	—	—	—	—	—	—	—	—
scarfbnum	1	21	21	0	3.3e-8	21	21	0	3.3e-8
scarfbnum	2	20	20	0	2.8e-9	20	20	0	2.8e-9
scarfbsum	1	20	20	0	3.6e-10	20	20	0	3.6e-10
scarfbsum	2	19	19	0	3.3e-10	19	19	0	3.3e-10
vonthmcp	1	37	38	1	1.5e-7	37	38	1	1.5e-7
vonthmge	1	—	—	—	—	—	—	—	—
duopoly	1	224	285	2	1.0e-11	161	274	3	6.6e-11
simple-ex	1	—	—	—	—	—	—	—	—
games	1	13	13	0	3.3e-8	13	13	0	3.3e-8
games	2	14	14	0	5.6e-8	14	14	0	5.6e-8
games	3	—	—	—	—	—	—	—	—
games	4	20	20	0	1.1e-10	20	20	0	1.1e-10
games	5	—	—	—	—	110	112	1	8.2e-7
games	6	22	22	0	2.4e-7	22	22	0	2.4e-7
games	7	24	24	0	9.0e-9	24	24	0	9.0e-9
games	8	17	17	0	1.2e-10	17	17	0	1.2e-10
games	9	23	23	0	2.0e-9	23	23	0	2.0e-9
games	10	18	18	0	1.5e-7	18	18	0	1.5e-7

thing we want to stress here about Table 1 is that the reader might get a wrong feeling about this basic solver since it fails on so many problems. However, we stress that the basic solver is in fact one of the best solvers which is currently available and that the test problems selected for this paper are just a subset of the most difficult problems from the GAMSLIB and MCPLIB collections. In fact, the overall behaviour of the basic solver is much better, and it is able to solve all other problems basically without any difficulties.

From the results in Table 2 we can deduce a couple of things: First of all, the global optimization technique usually does not become active if the basic method itself was able to solve the underlying problem. The only exception is problem `vonthmcp`. The fact that the tunneling methods became active for this problem, however, was just helpful in this case since the total number of iterations for the tunneling method is less than for the basic semismooth solver. We therefore believe that our switching criterion is quite useful also from a practical point of view. Furthermore, it does not seem to destroy the overall efficiency of the algorithm.

Second, we see that the tunneling methods are quite successful: While there are 14 failures in Table 1, there are only 3 failures left for both tunneling methods in Table 2. The failures occur on different test problems for the two tunneling versions: The classical tunneling function was not able to solve problems `duopoly` and `games` (third and fifth starting point) while the exponential tunneling method fails on the three starting points for problem `pgvon106`. We stress, however, that the exponential tunneling method was able to reduce the norm of the natural residual $r(x)$ to a value which almost satisfied the termination criterion, and that this happened for all three starting points of the `pgvon106` example. In view of our limited numerical results, it is therefore our feeling that the exponential tunneling method is slightly superior to the classical tunneling approach.

Finally, the results in Table 3 clearly indicate that the filled function methods are less successful than the tunneling methods. Both filled functions seem to have a similar behaviour, and they were able to solve four/five more problems than the basic semismooth solver, so the improvement is much worse than the one we obtained with the two tunneling approaches.

7 Final Remarks

In this paper we were interested in the development of more robust (and still efficient) solvers for the solution of mixed complementarity problems by using ideas from global optimization. To this end, we took a standard semismooth solver, presented a theoretically justified switching criterion and tested two global techniques (tunneling and filled functions) on a couple of very difficult test examples. The results indicate that especially the tunneling approach leads to substantial numerical improvements.

As part of our future research, we plan to investigate some further techniques within the algorithmic framework of this paper. In particular, we plan to study the influence of proximal-point and regularization methods as suitable solvers in Step (S.3) of Algorithm 3.7. Although neither the proximal-point nor the regularization methods are really global optimization techniques, they are sometimes quite helpful in improving the robustness of existing codes, see [1, 2, 26, 30].

References

[1] S.C. Billups: *Algorithms for complementarity problems and generalized equations.* Ph.D. Thesis, Computer Sciences Department, University of Wisconsin – Madison, Madison, WI, August 1995.

[2] S.C. Billups: *Improving the robustness of complementarity solvers using proximal perturbations.* Technical Report, Department of Mathematics, University of Colorado, Denver, CO, March 1996.

[3] S.C. Billups: *A homotopy based algorithm for mixed complementarity problems.* Technical Report, Department of Mathematics, University of Colorado, Denver, CO, February 1998.

[4] S.C. Billups and M.C. Ferris: *QPCOMP: A quadratic program based solver for mixed complementarity problems.* Mathematical Programming 76, 1997, pp. 533–562.

[5] A. Brooke, D. Kendrick and A. Meeraus: *GAMS: A User's Guide.* The Scientific Press, South San Francisco, CA, 1988.

[6] B. Chen, X. Chen and C. Kanzow: *A penalized Fischer-Burmeister NCP-function: Theoretical investigation and numerical results.* Preprint 126, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, September 1997 (revised May 1998).

[7] T. De Luca, F. Facchinei and C. Kanzow: *A semismooth equation approach to the solution of nonlinear complementarity problems.* Mathematical Programming 75, 1996, pp. 407–439.

[8] S.P. Dirkse and M.C. Ferris: *MCPLIB: A collection of nonlinear mixed complementarity problems.* Optimization Methods and Software 5, 1995, pp. 319–345.

[9] F. Facchinei and J. Soares: *A new merit function for nonlinear complementarity problems and a related algorithm.* SIAM Journal on Optimization 7, 1997, pp. 225–247.

[10] M.C. Ferris, C. Kanzow and T.S. Munson: *Feasible descent algorithms for mixed complementarity problems.* Mathematical Programming Technical Report 98-04, Computer Sciences Department, University of Wisconsin – Madison, Madison, WI, March 1998.

[11] M.C. Ferris and T.S. Munson: *Interfaces to PATH 3.0: Design, implementation and usage.* Computational Optimization and Applications, to appear.

[12] M.C. Ferris and J.-S. Pang: *Engineering and economic applications of complementarity problems.* SIAM Review 39, 1997, pp. 669–713.

[13] A. Fischer: *A special Newton-type optimization method.* Optimization 24, 1992, pp. 269–284.

[14] R. Fletcher: *Practical Methods of Optimization.* John Wiley & Sons, New York, NY, 1987.

[15] R. Ge: *A filled function method for finding a global minimizer of a function of several variables.* Mathematical Programming 46, 1990, pp. 191–204.

[16] C. Geiger and C. Kanzow: *On the resolution of monotone complementarity problems.* Computational Optimization and Applications 5, 1996, pp. 155–173.

[17] S. Gómez and C. Barrón: *The exponential tunneling method.* Technical Report, Instituto de Investigaciones en Matematicas Aplicadas y en Sistemas, Universidad Nacional Autonoma de Mexico, Mexico, July 1991.

[18] M.M. Kostreva and Q. Zheng: *Integral global optimization method for solution of nonlinear complementarity problems.* Journal of Global Optimization 5, 1994, pp. 181–193.

[19] A.V. Levy and S. Gómez: *The tunneling method applied to global optimization.* In: P.T. Boggs, R.H. Byrd and R.B. Schnabel (eds.): *Numerical Optimization.* SIAM, Philadelphia, PA, 1984.

[20] A.V. Levy and A. Montalvo: *The tunneling algorithm for the global minimization of functions.* SIAM Journal on Scientific and Statistical Computation 6, 1985, pp. 15–29.

[21] O.L. Mangasarian and M.V. Solodov: *Nonlinear complementarity as unconstrained and constrained minimization.* Mathematical Programming 62, 1993, pp. 277–297.

[22] L. Qi: *Convergence analysis of some algorithms for solving nonsmooth equations.* Mathematics of Operations Research 18, 1993, pp. 227–244.

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

[24] H. Sellami and S.M. Robinson: *Homotopies based on nonsmooth equations for solving nonlinear variational inequalities.* In: G. Di Pillo and F. Giannessi (eds.): *Nonlinear Optimization and Applications.* Plenum Press, New York, NY, 1996.

[25] H. Sellami and S.M. Robinson: *Implementation of a continuation method for normal maps.* Mathematical Programming 76, 1997, pp. 563–578.

[26] D. Sun: *A regularization Newton method for solving nonlinear complementarity problems.* Applied Mathematics and Optimization, to appear.

[27] D. Sun and R.S. Womersley: *A new unconstrained differentiable merit function for box constrained variational inequality problems and a damped Gauss-Newton method.* SIAM Journal on Optimization, to appear.

[28] L.T. Watson: *Solving the nonlinear complementarity problem by a homotopy method.* SIAM Journal on Control and Optimization 17, 1979, pp. 36–46.

[29] N. Yamashita and M. Fukushima: *On stationary points of the implicit Lagrangian for nonlinear complementarity problems.* Journal of Optimization Theory and Applications 84, 1995, pp. 653–663.

[30] G. Zhou, D. Sun and L. Qi: *Numerical experiments for a class of squared smoothing Newton methods for complementarity and variational inequality problems.* In: M. Fukushima and L. Qi (eds.): *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods.* Kluwer Academic Publishers, Norwell, MD, 1998, to appear.