A smoothing multidimensional filter method for nonlinear complementarity problems

Wu Congwei  Cao Jiping  Zhu Yahong
Xi'an Research Institute of High-tech  Xi'an Research Institute of High-tech  Xi'an Research Institute of High-tech
Tongxin Road No.2, 2102 Team, Baqiao District, Tongxin Road No.2, Baqiao District, Tongxin Road No.2, Baqiao District,
Xi'an City, Shaanxi Province, Xi'an City, Shaanxi Province, Xi'an City, Shaanxi Province,
P.R.China  P.R.China  P.R.China
Email:wucongweihome@sina.com  Email: xacjp123@163.com  Email: zyhbingchen@163.com

Abstract—A smoothing multidimensional filter method for solving NCP is proposed, and the solution of the complementarity problems on the framework of the filter-trust-region method is obtained. The new algorithm does not depend on any extra restoration procedure and the results of numerical experiments show its efficiency.

Keywords—Multidimensional filter techniques; Nonlinear complementarity problems; filter-trust-region method

I. INTRODUCTION

Let \( F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n \) be a continuous differentiable function. The nonlinear complementarity problem (NCP) is to find a vector \( x \in \mathbb{R}^n \) such that
\[
x \geq 0, \quad F(x) \geq 0, \quad x^T F(x) = 0,
\] (1)
For convenience, denote \( I = \{1, 2, ..., n\} \). Throughout this paper, \( \| \cdot \| \) denotes the Euclidean norm.

The traditional approach for NCP involves reformulating the problem as a constrained or unconstrained optimization problem. We discuss for solving this optimization based on the class of trust-region methods and also on that of multidimensional filter methods introduced by Gould and Sainvitu [1].

Definition 1 A function \( \phi : \mathbb{R}^2 \rightarrow \mathbb{R} \) is said to be an NCP function if it possesses the following characterization
\[
\phi(a, b) = 0 \Leftrightarrow a \geq 0, \quad b \geq 0 \quad \text{and} \quad ab = 0.
\]
In this paper, we will use Fischer-Burmeister functions \( \phi : \mathbb{R}^2 \rightarrow \mathbb{R} \) defined by
\[
\phi(a, b) = \sqrt{a^2 + b^2} - (a + b)
\]
It is obviously that \( \phi \) is not differentiable at zero point, which is difficult to get the global convergence results, so we use the following smooth approximation for the Fischer-Burmeister function,
\[
\phi_\mu(a, b) = \sqrt{a^2 + b^2 + \mu^2} - (a + b)
\] where \( \mu \geq 0 \) is a smooth parameter. Then NCP can be approximated by the following nonlinear least square problems,

\[
\min_{x \in \mathbb{R}^n} f_\mu(x) := \frac{1}{2} \Phi_\mu(x)^T \Phi_\mu(x)
\] (2)
where
\[
\Phi_\mu(x) = \begin{bmatrix}
\phi_\mu(x_1, F_1(x)) \\
\vdots \\
\phi_\mu(x_n, F_n(x))
\end{bmatrix}
\] (3)

Obviously, if the complementarity problem (1) is solvable, then the minimization problem (2) and (1) are equivalent when the parameter \( \mu \) tends to zero. Throughout this paper, to simplify notation we will use the abbreviations \( g_\mu(x) = \nabla_x f_\mu(x) \). Beside, \( g_{\mu,i}(x) \) denote the i-th component of \( g_\mu(x) \).

II. THE MULTIDIMENSIONAL FILTER METHOD

The projected gradient of the objective function \( f_\mu(x) \) into the feasible set of the problem (2) is defined componentwise by
\[
\overline{g}_\mu(x) = \begin{cases}
g_\mu(x), & x_i \geq g_\mu(x), \\
x_i, & x_i < g_\mu(x),
\end{cases}
\] (4)
where \( i \in I \).

Note that the problem (2) is a nonlinear optimization problem with a positive parameter \( \mu \), so we can use \( x^*_\mu \) denote the KKT point of the problem (2), and then we have the following Conclusion.

Lemma 1 For all \( \mu > 0 \), a point \( x^*_\mu \) is a KKT point for the problem (2) if and only if \( \overline{g}_\mu(x^*_\mu) = 0 \).

Lemma 2 Assume that \( f(x) \) is a twice-continuously differentiable function on a closed and bounded set, then the KKT point of (2) converges to the KKT point of the following problem
\[
\min_{x \in \mathbb{R}^n} f_\mu(x) := \frac{1}{2} \Phi_\mu(x)^T \Phi_\mu(x)
\] (5)
when the parameter \( \mu \) tends to zero.

To solve the optimization problem (2), we compute a trial step \( d_k \) at a given iterate \( x_k \) by the following trust-region
subproblem
\[ \min \ Q_k(d) = f_{R_k}(x_k) + \nabla f_{R_k}(x_k)^T d + \frac{1}{2} d^T B_k d, \]
\[ s.t. \quad x_k + d \geq 0, \]
\[ ||d|| \leq \Delta_k \]

Where \( \Delta_k \) is the trust region radius, \( B_k = \nabla \Phi_{R_k}(x_k) \nabla \Phi_{R_k}(x_k)^T \), and
\[ \nabla \Phi_{R_k}(x_k) = \begin{pmatrix} \frac{\partial \Phi_{R_k}(x)}{\partial x_1} & \cdots & \frac{\partial \Phi_{R_k}(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_{R_k}(x)}{\partial x_1} & \cdots & \frac{\partial \Phi_{R_k}(x)}{\partial x_n} \end{pmatrix} \]

The positive parameters \( \mu_i \) tends to zero during the iterate of algorithm. A trial point \( x_i^+ \) is then computed by the trial step \( d_i \), denote \( x_i^+ = x_i + d_i \).

In our context, Our aim is to encourage convergence to first-order critical points by driving every component of the projected gradient \( \bar{g}_{R_k}(x) = (\bar{g}_{R_k}(x), \bar{g}_{R_k}(x), \ldots, \bar{g}_{R_k}(x))^T \) to zero.

Definition 2 A iterate point \( x_i \) is said to dominate another point \( x_i \) if and only if \( ||g_{R_k}(x_i)|| \leq ||g_{R_k}(x_i)|| \), \( \forall i \in I \).

Definition 3 A filter set \( F \) is a set of points such that no pair dominates any other.

We say that a new trial point \( x_i^+ \) is acceptable for the filter \( F \) if and only if
\[ \forall x_i \in F, \exists j \in I \ | \ |g_{R_k}(x_i^+) - g_{R_k}(x_i)| < \gamma_e \ |\ g_{R_k}(x_i)||, \ (8) \]

Where \( \gamma_e = (0.1/\sqrt{n}) \).

If an iterate \( x_i \) is acceptable for the filter \( F \), we add it to the filter and remove from it every \( x_i \in F \), such that \( ||g_{R_k}(x_i^+) - g_{R_k}(x_i)|| \) for all \( i \in I \), i.e.
\[ F_{i+1} = F \cup \{x_i \ \{x_i \} \} \]
where,
\[ D_i = \{x_i \in F \ | \ |g_{R_k}(x_i^+) - g_{R_k}(x_i)|| \leq \gamma_e \ |\ g_{R_k}(x_i)||, \forall i \in I \} \] \( (10) \)

Note that the element of the set \( F \) is not \( x_i \), but \( (||g_{R_k}(x_i^+) - g_{R_k}(x_i)|| \leq \gamma_e \ |\ g_{R_k}(x_i)||, \forall i \in I \). when we say a new trial point \( x_i^+ \) is acceptable for the filter \( F \). Nevertheless, it is convenient to say that we add point \( x_i^+ \) into \( F \).

III. THE MULTIDIMENSIONAL FILTER ALGORITHM

At each step, after a subproblem is solved, the filter and traditional trust region criterion are all employed to determine whether to accept the trial point or not. In trust region criterion, if there is a good agreement between the model and the objective function value at the current trial point \( x_i^+ = x_i + d_i \), it is said to be successful iteration; otherwise, it is said to be unsuccessful iteration.

In following algorithm, the multidimensional filter criterion is a relaxation for the trust region criterion to a certain extent because we even accept the set of iterations which is not accepted by trust-region criterion but filter criterion.

Algorithm 1 The Multidimensional Filter Algorithm for NCP

Step 0: Initialization. An initial point and an initial trust-region radius \( \Delta_0 \) are Let an initial point \( x_0 \), an initial trust-region radius \( \Delta_0 > 0 \), and an initial filter set \( \gamma = (10^4, 10^8) \) be given, as well as constants \( \gamma_e = (0.1/\sqrt{n}) \), \( 0 < \gamma_1 < \gamma_2 < 1 < \gamma_3 \), \( 0 < \Delta_0 < \Delta_{\text{max}} \). Compute \( f_{R_k}(x_0), g_{R_k}(x_0), \gamma_{R_k}(x_0), B_0 \), set \( k_0 := 0 \).

Step 1: Test for optimality. If \( ||g_{R_k}(x_0)|| + \mu < e \), stop.

Step 2: Determine a trial step. Compute a solution \( d_i \) of the subproblem (6).

Step 3: If \( d_i = 0 \), set \( x_{i+1} = x_i \), \( \mu_{i+1} = 0 \), \( B_{i+1} = B_0 \), set \( k := k+1 \), and go to Step1; else set \( x_{i+1} = x_i + d_i \), and compute \( f_{R_k}(x_{i+1}), g_{R_k}(x_{i+1}) \).

Step 4: Test for optimality. If \( ||g_{R_k}(x_{i+1})|| + \mu_{i+1} < e \), stop; else, compute
\[ \rho_k = \frac{f_{R_k}(x_{i+1}) - f_{R_k}(x_{i})}{Q_k(0) - Q_k(d_i)} \] \( (11) \)

Step 5: Tests to accept the trial step.\[ \quad \text{If} \ \rho_k \geq \eta_k, \text{set} \ x_{i+1} = x_{i+1}, \ \Xi_k = \Xi_k \cup \{x_i \ \{x_i \} \} \] ;
\[ \quad \text{Else if} \ \rho_k < \eta_k \text{and} x_{i+1} \text{is acceptable for the filter} \ \Xi_k, \text{set} \ x_{i+1} = x_{i+1}, \ \Xi_k = \Xi_k \cup \{x_i \ \{x_i \} \} \ ;
\[ \quad \text{Otherwise, set} \ x_{i+1} = x_{i}, \ \Xi_k = \Xi_k . \]

Step 6: Update the trust-region radius and the smooth parameter.
\[ \Delta_{i+1} = \begin{cases} \min \{\Delta_{\text{max}}, \gamma_1 \Delta_i \}, & \text{if} \ \rho_k < \eta_k, \\ \min \{\Delta_{\text{max}}, \gamma_2 \Delta_i \}, & \text{if} \ \rho_k \in [\eta_1, \eta_2), \\ \mu_{i+1}, & \text{otherwise} \end{cases} \] \( (12) \)

\[ \mu_{i+1} = \begin{cases} \theta \mu_i, & \text{if} \ \mu_i > 0.1 \ |\ g_{R_k}(x_{i+1})||, \\ \mu_i, & \text{otherwise} \end{cases} \] \( (13) \)

Step 7: Compute \( f_{R_k}(x_{i+1}), g_{R_k}(x_{i+1}), \bar{g}_{R_k}(x_{i+1}) \), \( B_{i+1} \), set \( k := k+1 \), and go to step1.

It seems that we have to compute the value of the projected gradient and the objective function twice in every iteration. In practice, \( \mu_i \) seldom update because it is a sufficient small parameter. Thus, it is rarely to compute the value of the projected gradient and the objective function twice in every iteration. There is an advantage to choosing a large \( \Delta_{i+1} = \gamma_1 \Delta_i \) when \( \rho_k \geq \eta_k \), but it may be unwise to
choose it to be to large, so we give a upper bound $\Delta_{\text{max}}$, and set $\Delta_{k+1} = \min\{\Delta_{\text{max}}, \gamma_k \Delta_k\}$ when $\rho_k \geq \eta_2$.

Observe that the subproblem (6) is compatible during in algorithm, so we do not need any extra feasibility restoration phase in our algorithm, which differentiated our paper from those filter algorithm for nonlinear complementarity problems[2]. By the way, we are surprised to find that $B_{k+1} = \nabla \Phi_{\rho_k}(x_k) \nabla \Phi_{\rho_k}(x_k)^T$ can be computed easily instead of updating $B_{k+1}$ with higher numerical expenditure, because of (3) and (7).

IV. CONCLUSIONS AND NUMERICAL EXPERIMENTS

The new algorithm 1 presented in this paper combines with the multidimensional filter technique and the trust region method, which has a good numerical calculation results.

Now, we give some numerical results for the following 14 complementarity test problems in TABLE I. The values for the constants used in our tests are $\mu_0 = 10^{-1}$, $\varepsilon = 10^{-5}$, $\gamma = 10^{-2}$, $\gamma_1 = 0.25$, $\gamma_2 = 2$, $\eta_1 = 0.25$, $\eta_2 = 0.95$, $\theta = 0.1$, $\Delta_{\text{max}} = 10^3$, $\gamma = (10^3, ..., 10^3)^T$. The trust-region radius update is implemented as

$$
\Delta_{k+1} = \begin{cases} 
\gamma_k \Delta_k, & \text{if } \rho_k < \eta_1, \\
\Delta_k, & \text{if } \rho_k \in [\eta_1, \eta_2), \\
\min\{\Delta_{\text{max}}, \gamma_k \Delta_k\}, & \text{if } \rho_k \geq \eta_2.
\end{cases}
$$

Example 1.1 (Kojima-Shindo Nonlinear complementarity test problem) We choose the degenerate example in [2, 3].

Example 1.2 (Kojima-Shindo Nonlinear complementarity test problem) We choose the non-degenerate example in [3].

Example 2 (Kanzow Nonlinear complementarity test problem[2, 4]).

Example 3.1 We choose Example 2 with the constant $b = ((-1)^1, ..., (-1)^1)^T$ in [5].

Example 3.2 We choose Example 2 with the constant $b = ((-1)^1 \sqrt{1}, ..., (-1)^1 \sqrt{n})^T$ in [5].

Example 4 We choose Example 2 in [6].

The computational results are listed in Table 1, in which

$\text{iter}$ denotes the number of iterations, and $Resf$ stand for the computing accuracy, i.e. $Resf = f_{\rho_k}(x_k)$. The numerical results show that Algorithm 1 is robust and efficient. The number of iterations and computing accuracy for most problems are satisfactory.

<table>
<thead>
<tr>
<th>Example</th>
<th>Start point</th>
<th>iter</th>
<th>Resf</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>(0,0,0,0)$^T$</td>
<td>7</td>
<td>4.43e-12</td>
</tr>
<tr>
<td>1.1</td>
<td>(2,1,0.5,2)$^T$</td>
<td>9</td>
<td>1.02e-12</td>
</tr>
<tr>
<td>1.2</td>
<td>(0,0,0,0)$^T$</td>
<td>9</td>
<td>7.98e-15</td>
</tr>
<tr>
<td>1.2</td>
<td>(2,1,0.5,2)$^T$</td>
<td>8</td>
<td>2.84e-10</td>
</tr>
<tr>
<td>2</td>
<td>(3,2,1,2,3)$^T$</td>
<td>2</td>
<td>1.43e-13</td>
</tr>
<tr>
<td>2</td>
<td>(-2,...,-2)$^T$</td>
<td>3</td>
<td>9.17e-13</td>
</tr>
<tr>
<td>3.1</td>
<td>(2,...,2)$^T$</td>
<td>8</td>
<td>1.66e-11</td>
</tr>
<tr>
<td>3.2</td>
<td>(2,...,2)$^T$</td>
<td>5</td>
<td>1.05e-14</td>
</tr>
<tr>
<td>4</td>
<td>(10,1,0.5,10,10)$^T$</td>
<td>4</td>
<td>1.68e-11</td>
</tr>
<tr>
<td>4</td>
<td>(12,...,12)$^T$</td>
<td>9</td>
<td>5.84e-11</td>
</tr>
</tbody>
</table>

References