Journal of Computational Mathematics, Vol.21, No.2, 2003, 237–246.

A NONMONOTONE TRUST REGION ALGORITHM FOR NONLINEAR OPTIMIZATION SUBJECT TO GENERAL CONSTRAINTS*[1)]

Hongchao Zhang[†]
(*Department of Mathematics, University of Florida, Gainesville, FL, USA 32611*)

Abstract

In this paper we present a nonmonotone trust region algorithm for general nonlinear constrained optimization problems. The main idea of this paper is to combine Yuan's technique[1] with a nonmonotone method similar to Ke and Han [2]. This new algorithm may not only keep the robust properties of the algorithm given by Yuan, but also have some advantages led by the nonmonotone technique. Under very mild conditions, global convergence for the algorithm is given. Numerical experiments demonstrate the efficiency of the algorithm.

Key words: Nonlinear optimization, Nonmonotone algorithm, Trust region, General constraintss

1. Introduction

In this article, we consider the following nonlinear programming problem :

$$\min_{x\in\Re^n} \quad f(x) \tag{1.1}$$

$$s.t. \quad c_i(x) = 0, \qquad i = 1, 2, \ldots, m_e; \tag{1.2}$$

$$c_i(x) \geq 0, \qquad i = m_e + 1, \ldots, m, \tag{1.3}$$

where $f(x)$ and $c_i(x)$ $(i = 1, \ldots, m)$ are real functions defined in $\Re^n$, at least one of these functions is nonlinear, and $m \geq m_e$ are two non-negative integers.

In recent years, due to its nice results in real calculations, for example see [3], the nonmonotone trust region method has received many successful applications. One of the important reasons is that the nonmonotone method allows the sequence of iterates to follow the bottom of curved valleys (a common occurrence in difficult nonlinear problems) much more loosely. On the other hand, the monotonic reduction at every iteration, namely $f(x_{k+1}) < f(x_k)$, is not the intrinsic property to the convergence of the trust region method. Especially when the merit function is nondifferentiable, the nonsmoothness of the merit function may cause unnecessary reduction of the trust region bound (see [9]), a phenomenon similar to "Maratos effect". One technique to overcome this undesirable effect is the "second order step", but the price paid is that it must compute the value of the constraints at an auxiliary point and solve an additional subproblem. However the nonmonotone technique, like the watchdog technique, is a simple way which is helpful to overcome this difficulty. This is also one of our motivations to use the nonmonotone technique in our algorithm, as the merit function we use is nondifferentiable.

* Received July 10, 2000.

[1)] This work was done when the author was studying in the State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, P. O. Box 2719, Beijing 100080, China .

† Email address: hzhang@math.ufl.edu.

Up to now most nonmonotone trust region algorithms are applied in the unconstrained optimization, for example see [2] [5] [7], except Toint[3], which solves nonlinear optimization problems subject to convex constraints. In addition, Ke and Han [4] proposed a nonmonotone trust region method for equality constrained optimization problems based on a continuously differentiable merit function. In this paper, we try to present a nonmonotone trust region algorithm with a nondifferentiable merit function which can solve the general constrained optimization problems. The numerical results are also reported with this paper.

The paper is organized as follows. We present our algorithm in section 2 and give some preliminary results in section 3. Global convergence analysis of the algorithm are provided in section 4. Numerical results are reported in section 5. Conclusions are given in the last section.

2. The Algorithm

Define the L_∞ exact penalty function associated with (1.1)–(1.3)

$$P_{k,i}(x) = f(x) + \sigma_{k,i}\|c^-(x)\|_\infty, \tag{2.1}$$

where $\sigma_{k,i}$ is a penalty parameter and $c(x) = (c_1(x), \ldots, c_m(x))$, $c^-(x) \in \Re^m$ with

$$c_i^-(x) = c_i(x), \qquad i = 1, 2, \ldots, m_e; \tag{2.2}$$

$$c_i^-(x) = min(c_i(x), 0), \qquad i = m_e + 1, \ldots, m. \tag{2.3}$$

It is easy to see that $\|c^-(x)\|_\infty = 0$ if and only if x is a feasible point of (1.1)–(1.3). And under certain conditions, we can prove that the minimizer of the L_∞ penalty function is also a solution of the original nonlinear programming problem (1.1)–(1.3).

The subproblem we solve in our algorithm has the following form :

$$\min_{d\in\Re^n} \qquad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_{k,i}\|(c_k + A_k^T d)^-\|_\infty = \phi_{k,i}(d) \tag{2.4}$$

$$s.\ t. \qquad \|d\|_\infty \le \Delta_{k,i}, \tag{2.5}$$

where the superscript "-" has the same meaning as (2.2)–(2.3) and $A_k \in \Re^{n\times m}$ is the Jacobi matrix of the constraints. Assume that an inexact solution $s_{k,i}$ of (2.4)–(2.5) is computed such that it satisfies

$$\phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) \ge \tau\, \epsilon_{k,i} \min[\Delta_{k,i}, \epsilon_{k,i}/\|B_k\|_2], \tag{2.6}$$

where, $0 < \tau \le \frac{1}{2}$ is a constant, $\epsilon_{k,i} = \|g_k - A_k\lambda_{k,i}\|_\infty$ and $\lambda_{k,i} \in \Re^m$ is the Lagrange mutipliers at the current point x_k. Now we give our algorithm.

Algorithm 2.1 *(a nonmonotone trust region algorithm)*

Step 0 *Given* $x_0 \in \Re^n$, $\Delta_{0,0} > 0$, $\epsilon \ge 0$, $B_0 \in \Re^{n\times n}$ *symmetric;*
$\sigma_{0,0} > 0$, $\delta_{0,0} > 0$, $\bar{\epsilon} \ge 0$, *integer* $M \ge 0$, $M_0 = M$;
$1 \gg \eta > 0$, $P_{r(0,0)} = P_{0,0}(x_0)$, $k = i = 0$, $m(0) = 0$.

Step 1 *Solve the subproblem (2.4)-(2.5) for* $s_{k,i}$ *and at the same time compute* $\epsilon_{k,i}$.

Step 2 *If* $\epsilon_{k,i} \le \epsilon$ *and* $\|c^-(x_k)\|_\infty \le \epsilon$, *then stop.*
If $pred_{k,i} = \phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) \le \bar{\epsilon}$ *and* $\|c^-(x_k)\|_\infty \le \epsilon$, *then stop.*
If

$$pred_{k,i} < \sigma_{k,i}\, \delta_{k,i} \min[\Delta_{k,i}, \|c_k^-\|_\infty], \tag{2.7}$$

then

$$\sigma_{k,i+1} = 2\,\sigma_{k,i},\ \delta_{k,i+1} = \frac{1}{4}\,\delta_{k,i},\ \Delta_{k,i+1} = \Delta_{k,i},\ i := i + 1, \tag{2.8}$$

go to Step 1;

else *go to Step* 3.

Step 3 Let $P_{r(k,i)} = \max\limits_{0\le j\le m(k)}\{P_{k,i}(x_{k-j})\}$

$$m(k) = min\{m(k-1)+1, M_k\} \tag{2.9}$$

Compute $\rho_{k,i} = \frac{P_{r(k,i)} - P_{k,i}(x_k + s_{k,i})}{\phi_{k,i}(0) - \phi_{k,i}(s_{k,i})}$;

(i) if $\rho_{k,i} \ge \eta$, then $x_{k+1} = x_k + s_{k,i}$,

$$\Delta_{k+1} = \begin{cases} 2\,\Delta_k & if\ \rho_{k,i} \ge 0.9, \\ 1.5\,\Delta_k & if\ 0.9 > \rho_{k,i} \ge 0.5, \\ \Delta_k & \text{otherwise;} \end{cases} \tag{2.10}$$

$\sigma_{k+1,0} = \sigma_{k,i}$, $\delta_{k+1,0} = \delta_{k,i}$, $i := 0$, $k := k+1$, $M_{k+1} = M$; generate B_{k+1}.

(ii) if $0 < \rho_{k,i} < \eta$, then $\Delta_{k,i} = \Delta_{k,i}/4$.

(iii) if $\rho_{k,i} < 0$, then $\Delta_{k,i} = \Delta_{k,i}/4$, $M_k = 2\,M$.

Go to Step 1.

It is easy to see that our algorithm does not require the value of the merit function at x_{k+1} is lower than that at x_k. So the algorithm is a nonmonotone algorithm. M_k, a parameter for determining the reference value $P_{r(k,i)}$, is adjusted in the algorithm according to the ratio $\rho_{k,i}$. B_{k+1} is usually updated by adding a lower rank matrix, such as the BFGS method which only depends on the first order derivatives of the objective and that of the constraints.

3. Some Preliminary Results

In this section we make the following assumption:

Assumption 3.1. *$f(x)$ and $c_i(x)$, $i = 1, \ldots, m$ are twice continuously differentiable.*

Because three different stationary points, which were given by Yuan [1], are strongly associated with our algorithm, first we introduce them as follows:

Definition 3.2. *x^* is called a stationary point if*

1. $c^-(x^*) = 0$;
2. $d^T g(x^*) \ge 0$ *holds for all d satisfying*

$$d^T \nabla c_i(x^*) = 0, \quad (i = 1, \ldots, m_e);$$
$$d^T \nabla c_i(x^*) \ge 0, \quad (c_i(x^*) = 0, i = m_e+1, \ldots, m).$$

Definition 3.3. *x^* is called an infeasible stationary point if*

1. $\|c^-(x^*)\|_\infty > 0$;
2. $\min_{d\in\Re^n} \|(c(x^*) + A(x^*)^T\, d)^-\|_\infty = \|c^-(x^*)\|_\infty$.

Definition 3.4. *x^* is called a singular stationary point if*

1. $c^-(x^*) = 0$;
2. *there exists a sequence y_k converging to x^* such that $c^-(y_k) \ne 0$ and*

$$\lim_{k\to\infty} \min_{\|d\|_\infty \le \|c^-(y_k)\|_\infty} \frac{\|(c(y_k) + A(y_k)^T\, d)^-\|_\infty}{\|c^-(y_k)\|_\infty} = 1.$$

From the definitions above, a stationary point is also a Kuhn-Tucher point. The infeasible and singular stationary points are all Fritz John points. For more details of the properties of the three stationary points, one can see [1]. In the following lemmas we show that our algorithm will not cycle infinitely at an iterate unless it is an infeasible stationary point.

Lemma 3.5 *For any iteration point x_k, if x_k is not an infeasible stationary point, then cycle "step 1 $\to$ step 2 $\to$ step 1" must be finished in finite steps.*

Proof. If $\|c^-(x_k)\|_\infty = 0$, it is obvious that the lemma is true.

Now we assume that $\|c^-(x_k)\|_\infty > 0$. Because x_k is not an infeasible stationary point, there exists a constant $\mu_k > 0$ such that

$$\min_{\|d\|_\infty \le 1} \|(c_k + A_k^T\, d)^-\|_\infty = \|c_k^-\|_\infty - \mu_k. \tag{3.1}$$

Let d_k be a solution for

$$\|(c_k + A_k^T\, d_k)^-\|_\infty = \min_{\|d\|_\infty \le 1} \|(c_k + A_k^T\, d)^-\|_\infty. \tag{3.2}$$

From the convexity of $\|(c_k + A_k^T\, d)^-\|_\infty$ and the boundedness of B_k, we can get

$$\begin{aligned} \phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) &\ge \phi_{k,i}(0) - \phi_{k,i}(d_k\ \min[1, \frac{\Delta_{k,i}}{\|d_k\|_\infty}]) \\ &\ge \mu_k\, \sigma_{k,i}\ \min[1, \frac{\Delta_{k,i}}{\|d_k\|_\infty}] + O(\Delta_{k,i}). \end{aligned} \tag{3.3}$$

If the lemma is not true, $i \to \infty$ for some fixed k. We have $\lim_{i\to\infty} \sigma_{k,i} = \infty$ and $\Delta_{k,i}$ remains unchanged. Thus, there exists $N_k > 0$ and $\bar\mu_k > 0$ such that

$$\begin{aligned} \phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) &\ge \bar\mu_k\, \sigma_{k,i}\, \Delta_{k,i} \\ &\ge \bar\mu_k\, \sigma_{k,i}\ \min[\Delta_{k,i}, \|c_k^-\|_\infty] \end{aligned} \tag{3.4}$$

holds for all $i > N_k$. This contradicts (2.8) as $\delta_{k,i} \to 0$. so the lemma is true. □

By the above lemma, it is straightforward to see that for any fixed k if the Algorithm 2.1 does not stop in finite steps and $i \to \infty$, then x_k must be an infeasible stationary point.

Lemma 3.6 *For any iteration point x_k, if x_k is not an infeasible stationary point, then the innercycle "step 1 → step 2 → step 3 → ((ii) or (iii)) → step 1" must be finished in finite steps.*

Proof. If the innercycle does not terminate in finite steps, then $\rho^t_{k,i(t)} < \eta$, $\Delta^t_{k,i(t)} \to 0$ when $t \to \infty$, where t is the number of the cycles in the innercycle. For simplicity, we omit the subscripts $k, i(t)$ of P, ϕ, s, σ, ϵ, Δ, δ, ρ in the following analysis.

From the Algorithm 2.1, we know that

$$\frac{P(x_k) - P(x_k + s^t)}{\phi(x_k) - \phi(x_k + s^t)} \le \frac{P_{r(k,i(t))} - P(x_k + s^t)}{\phi(x_k) - \phi(x_k + s^t)} \le \eta < 1. \tag{3.5}$$

Similar to (3.4) in the Lemma 3.5, there exists a constant $N_k > 0$ such that $i(t) \le N_k$ when $t \to \infty$. So there exists $t_1 > 0$ such that $\sigma^t = \sigma^{t_1}$ and $\delta^t = \delta^{t_1}$ when $t \ge t_1$. In addition, $max\{\epsilon^t, \|c_k^-\|_\infty\} > \epsilon$, for the algorithm does not stop. So if $t > t^1$,

$$\begin{aligned} &\left|\frac{P(x_k) - P(x_k + s^t)}{\phi(x_k) - \phi(x_k + s^t)}\right| \\ \le\ &\left|\frac{\frac{1}{2}(s^t)^T(B_k - \nabla^2 f(\bar x))(s^t) + \sigma^t(\|(c_k + A_k^T s^t)^-\|_\infty - \|c^-(x_k + s^t)\|_\infty)}{\max\{\tau\epsilon^t \min[\Delta^t, \frac{\epsilon^t}{\|B_k\|_2}], \sigma^t\delta^t \min[\Delta^t, \|c_k^-\|_\infty]\}}\right| \\ \le\ &\frac{\frac{1}{2}\|B_k - \nabla^2 f(\bar x_k)\|_\infty \|s^t\|_\infty^2 + \sigma^{t_1} o(\|s^t\|_\infty)}{\bar\eta_k \min[\Delta^t, \bar\eta_k]}, \end{aligned} \tag{3.6}$$

where $\bar\eta_k$ is a constant and $\bar x_k = x_k + \theta s^t$ for some $0 < \theta < 1$. Because $\Delta^t \to 0$ when $t \to \infty$ and $\|s^t\|_\infty \le \Delta^t$,

$$\left|\frac{P(x_k) - P(x_k + s^t)}{\phi(x_k) - \phi(x_k + s^t)}\right| \to 1. \tag{3.7}$$

This contradicts (3.5). So the lemma is true.

4. Global Convergence

Throughout this section we make the following assumptions.

Assumption 4.1. 1. *$f(x)$ and $c_i(x)$, $i = 1, \ldots, m$ are twice continuously differentiable.*
2. *$\{x_k\}$ and $\{B_k\}$ are uniformly bounded. Suppose $\|B_k\|_2 \leq w$ for all k.*
The assumption on $\{x_k\}$ is common in analysis of the convergence of many algorithms, for in the case that $\{x_k\}$ is unbounded the original optimization may be ill-posed. However the boundedness of $\{B_k\}$ can be relaxed to be linearly increasing, that is $\|B_k\|_2 \leq T\,k$ for some constant T. We will discuss this condition at the end of the section. For the analysis of convergence we let $\epsilon = 0$ and $\bar{\epsilon} = 0$ in our algorithm.

By Lemma 3.6 we know that if all the iteration points are not infeasible stationary points, the number of updating penalty parameter at the k-th iteration, denoted by $N(k)$, must be a finite positive integer. From the Algorithm 2.1 we can see that $\sigma_{k,N(k)} = \sigma_{k+1,0}$.

Assumptoin 4.2. *We assume all the iteration points are not infeasible stationary points.*
Under the Assumption 4.2, if the algorithm does not stop after finite iterations we can give our following lemmas.

Lemma 4.3. *If Assumption is satisfied and the sequence of iteration points $\{x_k\}$ is bounded away from both infeasible stationary points and singular stationary points, there must exist two constants $0 < \mu < 1$ and $L > 0$ such that*

$$\min_{\|d\|_\infty \leq \|c^-(x_k)\|_\infty} \|(c_k + A_k^T\, d)^-\|_\infty \leq (1-\mu)(\|c_k^-\|_\infty) \tag{4.1}$$

holds for all $k \geq L$.

Proof. If (4.1) does not hold, there exists a subsequence $\{x_{k_i}\}$ such that

$$\min_{\|d\|_\infty \leq \|c^-(x_{k_i})\|_\infty} \|(c(x_{k_i}) + A^T(x_{k_i})\, d)^-\|_\infty > (1-\mu_i)(\|c^-(x_{k_i})\|_\infty \tag{4.2}$$

and

$$\|c^-(x_{k_i})\|_\infty > 0\,, \tag{4.3}$$

where μ_i is a number sequence converging to 0. By the boundedness of assumption on $\{x_k\}$ and the continuity of $\|c^-(x)\|_\infty$, $x \in \Re^n$, there must exist a constant L_1 such that $\|c^-(x_k)\|_\infty \leq L_1$ for all k. So there also exists a converging subsequence of $\{x_{k_i}\}$, also denoted by $\{x_{k_i}\}$, such that

$$\lim_{i\to\infty} x_{k_i} = \bar{x}, \qquad \lim_{i\to\infty} \|c^-(x_{k_i})\|_\infty = \|c^-(\bar{x})\|_\infty < \infty. \tag{4.4}$$

Hence, using (4.2) (4.3) and (4.4) we can prove

$$\lim_{i\to\infty} \min_{\|d\|_\infty \leq \|c^-(x_{k_i})\|_\infty} \frac{\|(c(x_{k_i}) + A^T(x_{k_i})\, d)^-\|_\infty}{\|c^-(x_{k_i})\|_\infty} = 1. \tag{4.5}$$

If $\|c^-(\bar{x})\|_\infty = 0$, by the defination of singular stationary point $\bar{x}$ is a singular stationary point.

If $\|c^-(\bar{x})\|_\infty > 0$, by the defination of infeasible stationary point and the convexity of $\|(c(\bar{x}) + A^T(\bar{x})\, d)^-\|_\infty$, we can get $\bar{x}$ is an infeasible stationary point.

However this contradicts the conditions of the lemma. So the lemma is true. □

Using Lemma 4.3, we can easily give the following result.

Lemma 4.4. *If $\lim_{k\to\infty} \sigma_{k,N(k)} = \infty$, then the sequence of iteration points $\{x_k\}$ is not bounded away from infeasible stationary points or singular stationary points.*

Proof. Assume the lemma were not true, by the Lemma 4.3, there must exist two constants $0 < \mu < 1$ and $L > 0$ such that (4.1) holds for all $k \geq L$. We define the vector d_k to be a solution for

$$\|(c(x_k) + A(x_k)^T\, d_k)^-\|_\infty = \min_{\|d\|_\infty \leq \|c^-(x_k)\|_\infty} \|(c(x_k) + A^T(x_k)\, d)^-\|_\infty. \tag{4.6}$$

Hence by the convexity of $\|(c(x)+A^T(x)\,d)^-\|_\infty$, the boundedness of $\{B_k\}$, the definition of $s_{k,i}$ and $\sigma_{k,i} \geq \sigma_{k-1,N(k-1)}$ for $0 \leq i \leq N(k)$, if $\|c^-(x_k)\|_\infty \neq 0$ we can get

$$\begin{aligned}
\phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) &\geq \phi_{k,i}(0) - \phi_{k,i}(d_k \ \min[1, \frac{\Delta_{k,i}}{\|d_k\|_\infty}]) \\
&\geq \mu\,\sigma_{k,i}\,\|c^-(x_k)\|_\infty \min[1, \frac{\Delta_{k,i}}{\|d_k\|_\infty}] + \\
&\quad O(\min[\|c^-(x_k)\|_\infty, \Delta_{k,i}]) \\
&\geq \mu\,\sigma_{k,i}\,\|c^-(x_k)\|_\infty \min[1, \frac{\Delta_{k,i}}{\|c^-(x_k)\|_\infty}] + \\
&\quad O(\min[\|c^-(x_k)\|_\infty, \Delta_{k,i}]) \\
&\geq \bar{\mu}\,\sigma_{k,i}\ \min[\|c^-(x_k)\|_\infty, \Delta_{k,i}] \qquad (4.7)
\end{aligned}$$

for all large k and $0 \leq i \leq N(k)$, $\bar{\mu}$ being a constant. If $\|c^-(x_k)\|_\infty = 0$, (4.7) can certainly be established. But (4.7) implies the boundedness of the penalty parameter. This contradicts the condition $\lim\limits_{k\to\infty} \sigma_{k,N(k)} = \infty$ in the lemma. So the lemma is true. □

From the Lemma 4.3 and Lemma 4.4 we can see that if the optimization problem has neither infeasible stationary points nor singular stationary points, the penalty parameter will remains bounded. In the following we will study the case the penalty parameter does not tend to infinity. The global convergence theorem states as follows :

Theorem 4.5. *If $\sigma_{k,i} = \sigma$ for all large k, then the sequence of iteration points $\{x_k\}$ is not bounded away from a K-T point.*

Proof. Without loss of generality, we can assume that $\sigma_{k,i} = \sigma_{k,0} = \sigma$, $\delta_{k,i} = \delta_{k,0} = \delta$, $P_{k,i}(x) = P_{k,0}(x) = P(x)$, and $i \equiv 0$ hold for all k. For simplicity, we also omit the subscript 0 of $\phi_{k,0}$, $\epsilon_{k,0}$, $\Delta^t_{k,0}$, $s^t_{k,0}$ where t stands for the t-th innercycle in the k-th iteration. Thus we can also assume

$$\phi_k(0) - \phi_k(s^t_k) \geq \max\{\tau\epsilon_k \min[\Delta^t_k, \frac{\epsilon_k}{\|B_k\|_2}], \sigma\delta \min[\Delta^t_k, \|c^-_k\|_\infty]\} \qquad (4.8)$$

holds for all k.

Now we will prove

$$\liminf_{k\to\infty} \max[\epsilon_k, \|c^-_k\|_\infty] = 0. \qquad (4.9)$$

If (4.9) does not hold, we can assume $\max[\epsilon_k, \|c^-_k\|_\infty] \geq \mu$ for all k, where $\mu > 0$ is a constant. Thus by (4.8) there exists constants $\mu_1 > 0$ and $T > 0$ such that

$$\begin{aligned}
\rho^t_{k,0} &= \frac{P_{r(k,0)} - P(x_k + s^t_k)}{\phi_k(0) - \phi_k(s^t_k)} \\
&\geq \frac{P(x_k) - P(x_k + s^t_k)}{\phi_k(0) - \phi_k(s^t_k)} \\
&\geq 1 - T\,\frac{o(\|s^t_k\|_\infty)}{\min[\Delta^t_k, \mu_1]}.
\end{aligned}$$

Therefore there exists a constant $\mu_2 > 0$ such that if $\Delta^t_k < \mu_2$ then $\rho^t_{k,0} > \eta$. Thus, by the technique of updating Δ^t_k, we can easily prove that

$$\Delta^t_k \geq \mu_3 > 0 \qquad (4.10)$$

for all k and $t \in [1, L(k)]$ where μ_3 is a constant and $L(k)$ is the total number of innercycles in the k-th iteration. So using (4.8) and the definition $x_k + s^{L(k)}_k = x_{k+1}$ we can get

$$\begin{aligned}
P_{r(k,0)} - P(x_k + s^{L(k)}_k) &= P_{r(k,0)} - P(x_{k+1}) \\
&\geq \eta \min\{\tau\mu \min[\mu_3, \frac{\mu}{w}], \sigma\delta \min[\mu_3, \mu]\} \\
&=: \tilde{\epsilon}. \qquad (4.11)
\end{aligned}$$

From (2.9) we know $m(k)+1 \geq m(k+1)$. Thus considering (4.11), for any k, we get

$$P_{r(k,0)} \geq P_{r(k+1,0)}. \tag{4.12}$$

Similar to (4.11) we can get

$$\begin{aligned} P_{r(k+1,0)} - P(x_{k+2}) &\geq \tilde{\epsilon} \\ \vdots \quad \vdots &\quad \vdots \\ P_{r(k+2\,M,0)} - P(x_{k+2M+1}) &\geq \tilde{\epsilon}. \end{aligned} \tag{4.13}$$

From (4.12) (4.13) we have that

$$P_{r(k,0)} \geq \max\{P(x_{k+1}), P(x_{k+2}), \ldots, P(x_{k+2M+1})\} + \tilde{\epsilon}. \tag{4.14}$$

Using (2.9) and the above equation we know

$$P_{r(k,0)} \geq P_{r(k+2M+1,0)} + \tilde{\epsilon}. \tag{4.15}$$

But (4.15) contradicts the boundedness of $P(x)$. So (4.9) holds. Therefore using the boundedness of $\{x_k\}$ there exits a subsequence of $\{x_k\}$ converging to a K-T point. Thus the the theorem is true. □

Now we consider the relaxation for the boundedness of $\{B_k\}$. We know for many updating techniques of B_k,

$$\|B_k\|_2 \leq T\,k \tag{4.16}$$

can be established for some constant T. In the rest of the section we assume B_k is updated from iteration to iteration such that (4.16) is satisfied. In ths case, however, we should modify our algorithm slightly, replacing (2.7) by

$$pred_{k,i} < \sigma_{k,i}\,\delta_{k,i}\,\min[\Delta_{k,i}, \frac{1}{k}, \|c_k^-\|_\infty]. \tag{4.17}$$

When we consider the preliminary results in section 2, k is a fixed number. So the Lemma 3.5 and 3.6 are still valid. It can also be seen that the Lemma 4.3 is independent of $\|B_k\|$ and independent of (2.7). Hence Lemma 4.3 remains valid. As for Lemma 4.4, if it does not hold, similar to (4.7) we can prove

$$\begin{aligned} \phi_{k,i}(0) - \phi_{k,i}(s_{k,i}) &\geq \phi_{k,i}(0) - \phi_{k,i}(d_k\ \min[1, \frac{\min[\Delta_{k,i}, 1/k]}{\|d_k\|_\infty}]) \\ &\geq \mu\,\sigma_{k,i}\,\|c^-(x_k)\|_\infty \min[1, \frac{\min[\Delta_{k,i}, 1/k]}{\|c^-(x_k)\|_\infty}] + \\ &\quad O(\min[\|c^-(x_k)\|_\infty, 1/k, \Delta_{k,i}]) \\ &\geq \bar{\mu}\,\sigma_{k,i}\ \min[\|c^-(x_k)\|_\infty, \Delta_{k,i}, 1/k], \end{aligned} \tag{4.18}$$

which gives a contradiction. As for Theorem 4.5, if it does not hold, similar to (4.10) we can prove

$$\Delta_k^t \geq \frac{\mu_3}{k \log k} \tag{4.19}$$

for all k and $t \in [1, L(k)]$ where μ_3 is a constant. Similar to (4.15) we can get

$$P_{r(k,0)} \geq P_{r(k+2M+1,0)} + \frac{\tilde{\epsilon}}{(k+2M)\log(k+2M)}, \tag{4.20}$$

where $\tilde{\epsilon}$ is another constant. So we also get a contradiction. From the reason above, we can re-establish Lemma 4.4 and Theorem 4.5.

5. Numerical Results

A FORTRAN subroutine was programmed to test our algorithm. Our experiments were done in double precision arithmatic on an indigo workstation at the State Key Laboratory of Scientific and Engineering Computing.

As in Yuan [1] we also solve the nonsmooth subproblem (2.4)– (2.5) by reformulating it as the following equivalent quadratic program :

$$\min \quad \bar{g}_{k,i}^T \bar{d} + \frac{1}{2} \bar{d}^T \bar{B}_k \bar{d}^T \tag{5.1}$$

subject to

$$\begin{aligned} d_{n+1} + (c_i(x_k) + d^T \nabla c_i(x_k)) &\geq 0, \quad i = 1, \ldots, m \\ d_{n+1} - (c_i(x_k) + d^T \nabla c_i(x_k)) &\geq 0, \quad i = 1, \ldots, m_e \\ -\Delta_{k,i} \leq d_i &\leq \Delta_{k,i}, \quad i = 1, \ldots, n \\ d_{n+1} &\geq 0, \end{aligned}$$

where $d = (d_1, \ldots, d_n)^T, \bar{d} = (d^T, d_{n+1})^T, \bar{g}_{k,i}^T = (g_k^T, \sigma_{k,i})$, and $\bar{B}_k$ is the $(n+1) \times (n+1)$ matrix that is defined by

$$\bar{B}_k = \begin{bmatrix} B_k & 0 \\ 0 & 0 \end{bmatrix}.$$

We solve the quadratic programming subproblems by Flecher's Harvell subroutine VE02AD.

The test problems are from Hock and Schittkowski. For each problem, the standard initial point is used. We choose initial parameters $M = 2$, $\Delta_{0,0} = 10$, $\sigma_{0,0} = 10$, $\delta_{0,0} = 0.01, \eta = 0.1$, $\epsilon = 10^{-8}$, $\bar{\epsilon} = 10^{-10}$. We only update B_k at acceptable points. The initial Lagrangian Hessian estimating B_0 is I and B_k is updated by the Powell's safeguarded BFGS formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \tag{5.2}$$

where

$$y_k = \begin{cases} \hat{y}_k, & \text{if } \hat{y}_k^T s_k \geq 0.1\, s_k^T B_k s_k, \\ \theta_k \hat{y}_k + (1 - \theta_k) B_k s_k, & \text{otherwise,} \end{cases} \tag{5.3}$$

and $\hat{y}_k = g_{k+1} - g_k + (\nabla c(x_{k+1}) - \nabla c(x_k))\, \lambda_{k,N(k)}$, $s_k = x_{k+1} - x_k$, $\theta_k = 0.9 s_k^T B_k s_k / (s_k^T B_k s_k - s_k^T \hat{y_k})$, $\lambda_{k,N(k)}$ is a multiplier associated with (2.4)–(2.5).

The test problems are also solved by Powell's subroutine VMCWD, which is a very successful algorithm and uses watch-dog techniques. The error tolerance for VMCWD is 10^{-8}. For comparison, we let $M = 0$ in our algorithm to make the algorithm become a monotone one and solve the algorithm again. The numerical results are listed in Table 1(see the Appendix). In the table, for example, "No.43" means problem 43 in Hock and Schittkowski [11]. (F-G) means the numbers of function and gradient calculations respectively. "Total" is the total number of calculations which means Total $= (1 + m) \times$ F $+ n \times (m + 1) \times$ G. "Res" is the infinite norm of the residual for the K-T condition at the computed solution $\tilde{x}$ which means Res $= \|c(\tilde{x})^-\|_\infty + \|g(\tilde{x}) - A(\tilde{x})^T \lambda(\tilde{x})\|_\infty$, where $\lambda(\tilde{x})$ is an approximate Langrange multiplier for the subproblem at the solution.

From the numerical results we can see that when the scale of the problem is small, our nonmonotone algorithm is comparable to VMCWD and the monotone one. But when the scale of the problem is a little larger and the condition is more complex, the numerical results show that our nonmonotone algorithm is better than the other two algorithms. However in real computation our algorithm seems to have to solve a little more subproblems. And how to solve the subproblem (2.4)–(2.5) efficiently is another important matter in real computation. When we uses VE02AD to solve the subproblem, in some iterations VE02AD failed to provide accurate solutions for the subproblem. For example, due to this reason the monotone algorithm failed to solve the problem No.27. What's more, it is important to select the parameter M. We have tested the condition when $M = 1, M = 3, M = 4$. Numerical results show that both a large M and a small M will cause the algorithm become worse. We know at least M and at most $2M$ recent merit function values will be used for selecting the k-th reference value. So if we have computed a very "good" iteration point in the recent M or $2M$ iterations, namely the

merit function value at this point is very "low" compared with other points, it may be discarded away and hasn't any contribution on the following iterations. And we observed that this case often occured in real computation when M is large. However, if M is small, the algorithm can not make full use of the advantages of the nonmonotone technique. Therefore finally we select $M = 2$ in our algorithm.

6. Conclusions and Discussions

In this paper we present a nonmonotone trust region algorithm for general constraints and give its global convergence. The numerical results show that the nonmonotone technique is really helpful to solve difficult nonlinear problems. We think it should be nteresting to combine the nonmonotone and "second order step" techniques in trust region method when the merit function is nonsmooth. And the similar case in line search method have been studied in [10]. How to select an efficient parameter M is another important problem in real computations which still merits further studies. In general, nonmonotone trust region method is still in infancy and a lot of questions need to be solved.

Acknowledgements. The author would like to thank professor Ya-xiang Yuan very much, who carefully directed me to write this paper. Thanks are also due to two anonymous referees, whose comments and suggestions greatly improved this paper.

Appendix

Table 1

	VMCWD		Our Algorithm (M=0)		Our Algorithm (M=2)	
No. (n-m)	Total (F-G)	Res	Total (F-G)	Res	Total (F-G)	Res
7 (2-1)	84 (14-14)	4.27E-9	62 (11-10)	7.74E-9	66 (11-11)	3.22E-8
12 (2-1)	60 (10-10)	2.19E-9	90 (19-13)	1.20E-7	68 (14-10)	2.21e-13
22 (2-2)	63 (7-7)	4.78E-12	45 (5-5)	3.56E-13	45 (5-5)	3.56E-13
27 (3-1)	504 (63-63)	5.82E-5	fail	-	114 (15-14)	1.49E-5
35 (3-1)	64 (8-8)	2.77E-6	54 (9-6)	3.95E-7	52 (8-6)	1.31E-8
43 (4-3)	300 (15-15)	2.24E-6	256 (16-12)	6.10E-8	260 (17-12)	1.63E-7
48 (5-2)	180 (10-10)	8.44E-6	189 (13-10)	1.78E-8	174 (13-9)	1.47E-5
66 (3-2)	84 (7-7)	1.10E-6	672 (68-52)	8.54E-6	114 (11-9)	4.31E-9
100 (7-4)	1000 (25-25)	2.79E-5	1170 (38-28)	2.69E-7	760 (26-18)	3.24E-6
113 (10-8)	1683 (17-17)	2.79E-5	14094 (166-140)	8.37E-5	1818 (22-18)	2.47E-8
285 (15-10)	13024 (74-74)	5.93E-6	8096 (61-45)	3.89E-9	4895 (40-27)	3.82E-7
384 (15-10)	13904 (79-79)	7.77E-5	11088 (78-62)	2.85E-6	5236 (41-29)	1.00E-7
385 (15-10)	12486 (71-71)	3.24E-5	20020 (140-112)	6.17E-4	6347 (52-35)	7.70E-8
386 (15-11)	19008 (99-99)	4.51E-5	50484 (292-261)	4.60E-4	6696 (48-34)	8.93E-8
387 (15-11)	17664 (92-92)	2.62E-4	17400 (115-89)	5.10E-4	5124 (37-26)	1.38E-6
388 (15-15)	16280 (55-55)	2.14E-6	42976 (196-166)	6.13E-4	6560 (35-25)	3.48E-8
389 (15-15)	15688 (53-53)	3.60E-6	47104 (214-182)	1.93E-4	6512 (32-25)	1.04E-8

References

[1] Y. Yuan, "On the convergence of a new trust region algorithm", *Numerische Mathematik*, **70**(1995), 515-539.

[2] X.Ke and J.Han," Global convergence of a class of new trust region algorithms",*Acta Mathematicae Applicatae Sinica*, **18**:4(1995), 608-615.

[3] Ph.L. Toint, "A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints", *Math. Prog.,* **77**(1997), 69-94.

[4] X.Ke and J.Han, "A nonmonotone trust region algorithm for equality constrained optimization", *Science in China (Series A)*, **38**(1995), 683-695

[5] N.Y. Deng, Y. Xiao and F.J. Zhou, "Nonmonotonic trust region algorithm", *JOTA*, **26**(1993), 259-285.

[6] Y.H.Dai,"On nonmonotone line search", Report ICM-2000-003, ICMSEC, Academia Sinica, 2000.

[7] Z.F.Li and N.Y. Deng, "A new family of nonmonotonic trust region algorithms and its properties", *Acta Mathematicae Applicatae Sinica*, **22**:3(1999), 457-465.

[8] Y. Yuan, "On the superlinear convergence of a trust region algorithm for nonsmooth optimization", *Math. Prog.,* **31**(1985), 269-285.

[9] Y.Yuan, "An example of only linearly convergence of trust region algorithms for nonsmooth optimization", *IMA J. Numer. Anal.,* **4**(1984), 327-335.

[10] E.Panier and A.Tits, "Avoiding the Maratos effect by means of a nonmonotone line search I: general constrained problems", *Siam Journal on Numerical Analysis,* **28**(1991), 1183-1195.

[11] W. Hock and K. Schittkowski, "Test examples for nonlinear programming codes" Lecture Notes in Eco. and Math. Systems 187, 1981.