J Math Model Algor (2011) 10:109–118
DOI 10.1007/s10852-010-9143-z

Global Convergence of a Nonmonotone Trust Region Algorithm with Memory for Unconstrained Optimization

Zhensheng Yu · Anqi Wang

Received: 14 January 2009 / Accepted: 21 June 2010 / Published online: 9 July 2010

Abstract In this paper, we consider a trust region algorithm for unconstrained optimization problems. Unlike the traditional memoryless trust region methods, our trust region model includes memory of the past iteration, which makes the algorithm less myopic in the sense that its behavior is not completely dominated by the local nature of the objective function, but rather by a more global view. The global convergence is established by using a nonmonotone technique. The numerical tests are also given to show the efficiency of our proposed method.

Keywords Trust region · Memory model · Nonmonotone technique · Global convergence

Mathematics Subject Classifications (2010) 65K05 · 90C30

1 Introduction

In this paper, we consider the following unconstrained optimization problem:

$$\min \ f(x) \ \text{ s.t. } \ x \in R^n, \tag{1}$$

where $f(x)$ is a twice continuously differentiable function from R^n to R.

Trust region method is one of the most well-known method for solving problem (1). Due to its strong convergence and robustness, trust region methods have been proved to be efficient for solving problem (2), and there are many researches on trust region methods available for solving such problem, see, for example, [1–5].

This work was supported by Innovation Program of Shanghai Municipal Education Commission (No. 10YZ99) and Shanghai Leading Academic Discipline Project (No. S30501).

Z. Yu (✉) · A. Wang
College of Science, University of Shanghai for Science and Technology, Shanghai, 200093, People's Republic of China
e-mail: zhsh-yu@163.com

For a given iterate point, x_k, in traditional trust region methods, the trial step is usually obtained by solving the following trust region subproblem:

$$\begin{aligned} &\min \ \langle g_k, d\rangle + \tfrac{1}{2}\langle d, H_k d\rangle, \\ &\text{s.t} \ \ \|d\| \le \Delta_k, \end{aligned} \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product, $g_k = \nabla f(x_k)$ and H_k is a symmetric matrix that approximates the Hessian matrix of $f(x_k)$, $\Delta_k > 0$ is the trust region radius.

The trust region subproblem (2) shows that the trial step depends purely on the local information of the objective function. In other words, the trust region model based solely on the first and the second derivatives of the objective function at current point, x_k; such a model is called a memoryless model.

It can be seen that the pure local nature of the trust region iteration is detrimental when the objective function is very nonlinear, in the sense that the second-order Taylor series varies quickly as a function of x. For instance, this is the case when the function has local ripples which have little global effect on the shape of the objective. A memoryless iteration may then be fooled by the local nature of the function and may easily loose track of the more global picture, although the latter is crucial for determining search directions that will enable substantial progress of the algorithm.

To overcome this drawback, we can remember the model we have seen in previous iterations and mix the information with the local model in the current iteration; such a model is called a memory model.

The first memory model was proposed by N.I.M Gould et al. for unconstrained optimization with a line search algorithm (see [6]). In [5], by combining with a conditional model, Conn et al. analyzed the global convergence of trust region algorithm with memory, but no numerical tests are presented.

The nonmonotone technique was originally proposed by Grippo et al. [7] for unconstrained optimization based on Newton's method. Numerical tests showed that the nonmonotone technique are helpful to overcome the case where the sequence of iterates follows the bottom of curved narrow valleys, a common occurrence in difficult nonlinear problems. In the past few decades, many nonmonotone trust region algorithms have been proposed to solve unconstrained and constrained optimization [8–11]. Since, in general, the trial step obtained by the memory model may not be a decrease direction, in this paper, we consider combining the nonmonotone trust region technique proposed by Toint [10] with the memory model to generate the iterate sequence. We establish the global convergence of the proposed algorithm and present numerical tests to show the efficiency of the algorithm.

This paper is organized as follows: In Section 2, we introduce our trust region algorithm with memory. In Section 3, we discuss the global convergence of the proposed algorithm. The conclusion with final remarks is presented in Section 4.

2 Algorithm

In this section, we describe the trust region algorithm with memory.

For a current point x_k, we denote the local model by

$$m_k(x) = f(x_k) + \langle g_k, x - x_k\rangle + \frac{1}{2}\langle x - x_k, H_k(x - x_k)\rangle, \tag{3}$$

Following [5] and [6], we consider the memory model

$$m_k^M(x) = (1-\mu_k)m_k(x) + \mu_k m_{k-1}^M(x), \tag{4}$$

where $\mu_k \in [0,1)$ is a parameter. In this paper, we choose $\mu_k = \min\{\overline{\mu}, \|x_k - x_{k-1}\|, \Delta_k\}$ for some $\overline{\mu} \in (0,1)$. Furthermore, we set $\mu_0 = 0$ since there is nothing to be remembered at the first iteration.

Let

$$g_k^M = \nabla m_k^M(x_k), \quad H_k^M = \nabla_{xx}^2 m_k^M(x_k), \tag{5}$$

The trust region subproblem with memory is defined as follows:

$$\begin{aligned} &\min \ \langle g_k^M, d\rangle + \tfrac{1}{2}\langle d, H_k^M d\rangle, \\ &\text{s.t} \ \ \|d\| \le \Delta_k. \end{aligned} \tag{6}$$

Lemma 1 [7] *Let d_k be a solution of (6). Then we have*

$$-\langle g_k^M, d\rangle - \frac{1}{2}\langle d, H_k^M d\rangle \ge \frac{1}{2}\|g_k^M\| \min\{\Delta_k, \frac{\|g_k^M\|}{\|H_k^M\|}\}. \tag{7}$$

In general, the search direction generated by (6) is a descent direction for $m_k^M(x)$ but may not be a descent direction for $f(x)$. Therefore, a natural choice is to employ the nonmonotone technique to determine whether the trial step can be accepted. In what follows, we introduce the nonmonotone trust region technique, see [10] for example.

Define

$$Pred_k = m_k^M(x_k) - m_k^M(x_k + d_k), \tag{8}$$

and compute

$$f_{l(k)} = \max_{i=k-m(k),\cdots,k} f_i, \tag{9}$$

where $m(k) = \min\{m(k-1)+1, N, N_k\}$, N is a given positive constant integer, N_k is a variable positive integer and f_i denotes $f(x_i)$.

Furthermore we compute

$$\rho_{1,k} = \frac{f_{l(k)} - f(x_k + d_k)}{\Sigma_{i=l(k)}^k Pred_i}, \tag{10}$$

$$\rho_{2,k} = \frac{f_k - f(x_k + d_k)}{Pred_k}, \tag{11}$$

$$\rho_k = \max\{\rho_{1,k}, \rho_{2,k}\}. \tag{12}$$

The nonmonotone trust region algorithm with memory is now stated formally as follows:

Algorithm 2.1

Step 0: Choose $x_0 \in R^n$, $H_0 \in R^{n\times n}$ symmetric, $g_0 = \nabla f_0, \overline{\mu} \in (0,1), \eta \in (0,1)$, $m(-1) = 0$, $N > 0$, $N_0 > 0$, $\Delta_{\min} > 0$, $\mu_0 = 0$, $k := 0$.
Step 1: If $g_k = 0$ stop.
Step 2: Set $\Delta = \Delta_{\min}$.
Step 3: Define model m_k^M and compute g_k^M and H_k^M; if $g_k^M = 0$, set $g_k^M = g_k$.
Step 4: Compute d_k, $Pred_k$, $\rho_{1,k}$, $\rho_{2,k}$, ρ_k.
Step 5: If $0 < \rho_k < \eta$, set $\Delta := \frac{1}{2}\Delta$, $N_k = N$ goto Step 3.
If $\rho_k \leq 0$, set $\Delta := \frac{1}{2}\Delta$, $N_k := N_k + 1$, goto Step 3.
Step 6: If $\rho_k \geq \eta$, set $\Delta_k = \Delta$, $x_{k+1} = x_k + d_k$, $N_{k+1} = N_k$. update H_k as H_{k+1}, $k := k+1$ goto Step 1.

In the above algorithm, we call Steps 5-3-4-5 interior cycle.

3 Global Convergence

In this paper, we assume that the algorithm generates finitely many iterations and make the following assumptions.

A1 The generated points, $\{x_k\}$, are contained in a closed convex set Ω.
A2 For any $x \in \Omega$, $\nabla m_i(x)$ and H_i are uniformly bounded.
A3 $f(x)$ is bounded below on Ω.

We first give some basic lemmas about the model $m_k^M(x)$.

Lemma 2 [5] *If the model is defined by* (4). *Then for each k, we have*

$$m_k^M(x) = \sum_{i=0}^{k}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j)m_i(x). \tag{13}$$

Moreover,

$$\sum_{i=0}^{k}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j) \leq \frac{1}{1-\overline{\mu}}. \tag{14}$$

Lemma 3 *If Assumptions* **A1** *and* **A2** *hold then there exist positive constants* k_1 *and* k_2 *such that*

$$\|g_k^M - g_k\| \leq k_1\Delta_k, \tag{15}$$

$$\|H_k^M - H_k\| \leq k_2\Delta_k. \tag{16}$$

Proof From (13), we have

$$g_k^M = \nabla m_k^M(x_k) = \sum_{i=0}^{k}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j)\nabla m_i(x_k), \tag{17}$$

and

$$H_k^M = \nabla m_{xx}^2 M_k(x_k) = \sum_{i=0}^{k}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j)\nabla_{xx}^2 m_i(x_k). \tag{18}$$

Thus for all k, we have

$$\begin{aligned} g_k^M - g_k &= \sum_{i=0}^{k-1}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j)g_i + (1-\mu_k)g_k - g_k \\ &= \sum_{i=0}^{k-1}(1-\mu_i)(\Pi_{j=i+1}^{k}\mu_j)g_i - \mu_k g_k. \end{aligned} \tag{19}$$

From Assumptions **A1** and **A2**, there exist two positive constants b_1, b_2 such that

$$\|\nabla m_i(x_k)\| \le b_1, \ \|H_i\| \le b_2, \text{for } i = 1, 2, \cdots, k. \tag{20}$$

Define

$$\theta_k = \sum_{i=0}^{k-1}(1-\mu_i)(\Pi_{j=i+1}^{k-1}\mu_j),$$

then from Lemma 2, we have

$$\theta_k \le \frac{1}{1-\overline{\mu}},$$

and hence, from the definition of μ_k, we have

$$\begin{aligned} \|g_k^M - g_k\| &\le \mu_k b_1(\theta_k + 1) \\ &\le b_1(\theta_k + 1)\mu_k \\ &\le \frac{b_1(2-\overline{\mu})}{1-\overline{\mu}}\Delta_k. \end{aligned}$$

Let $k_1 = \frac{b_1(2-\overline{\mu})}{1-\overline{\mu}}$, we get (15). Similarly, we have (16). □

The next result shows that the algorithm is well defined.

Lemma 4 *Under Assumptions* **A1** *and* **A2***, the algorithm is well defined, that is, the interior cycle Steps* 5-3-4-5 *must stop after finitely many iterations.*

Proof We prove the conclusion by contradiction, denote the *ith* iteration in iterate k by k, i, and the corresponding values denoted by $\Delta_{k,i}$, $Pred_{k,i}$ etc. If the conclusion is not true, then

$$\rho_{k,i} < \eta, \quad \lim_{i\to\infty} \Delta_{k,i} = 0. \tag{21}$$

Since $\|g_{k,i}^M\| \neq 0$, we have for i large enough

$$
\begin{aligned}
|\rho_{2,(k,i)} - 1| &= \left|\frac{f_k - f(x_k + d_{k,i})}{Pred_{k,i}} - 1\right| \\
&= \left|\frac{\langle g_k - g_{k,i}^M, d_{k,i}\rangle + \frac{1}{2}\langle d_{k,i}, (\nabla^2 f(x_k) - H_{k,i}^M)d_{k,i}\rangle + o(\|d_{k,i}\|^2)}{Pred_{k,i}}\right| \\
&\leq \frac{1}{Pred_{k,i}}\left[\|g_k - g_{k,i}^M\|\|d_{k,i}\| + \frac{1}{2}(\|\nabla^2 f(x_k) H_{k,i}^M\|)\|d_{k,i}\|^2 + o(\|d_{k,i}\|)^2\right] \\
&\leq \frac{\|g_k - g_{k,i}^M\|\|d_{k,i}\| + \frac{1}{2}(\|\nabla^2 f(x_k) - H_{k,i}^M\|)\|d_{k,i}\|^2 + o(\|d_{k,i}\|)^2}{\frac{1}{2}\|g_{k,i}^M\| \min\left\{\Delta_{k,i}, \frac{\|g_{k,i}^M\|}{\|H_{k,i}^M\|}\right\}} \\
&\leq \frac{k_1 \Delta_{k,i}\|d_{k,i}\| + \gamma_k\|d_{k,i}\|^2 + o(\|d_{k,i}\|)^2}{\frac{1}{2}\|g_{k,i}^M\|\Delta_{k,i}} \\
&\to 0 (as\ i \to \infty).
\end{aligned}
$$

where $\gamma_k = \frac{1}{2}(\sup_{x\in\Omega} \|\nabla^2 f(x)\| + \frac{b_2}{1-\overline{\mu}}) > 0$. Therefore, we have $\rho_{k,i} = \max\{\rho_{1,k,i}, \rho_{2,\ k,i}\} \geq \eta$, which contradicts (21). This completes the proof. □

For every k, we define the reference iteration associated with k by

$$
r(k) = \begin{cases} l(k), & \text{if } \rho_k = \rho_{1,k}, \\ k, & \text{if } \rho_k = \rho_{2,k}. \end{cases}
$$

Lemma 5 [8] *For each $k \geq 1$, we have*

$$
f(x_{r(k)}) - f(x_{k+1}) \geq \frac{1}{2}\eta \sum_{j=r(k)}^{k} \|g_j^M\| \min\left\{\Delta_j, \frac{\|g_j^M\|}{\|H_j^M\|}\right\}.
$$

Theorem 1 *Under the Assumptions* **A1–A3**, *we have*

$$
\liminf_{k\to\infty} \|g_k^M\| = 0. \tag{22}
$$

Proof Similar to Theorem 1 in [9], we know the sequence $\{f_{l(k)}\}$ is non increasing, and by Assumption **A1**, it is convergent.

We prove (22) by contradiction, assume that there exists a $\varepsilon > 0$ such that for all k, $\|g_k^M\| \geq \varepsilon$. We first prove that

$$
\liminf_{k\to\infty} \Delta_k = \widehat{\Delta} > 0. \tag{23}
$$

Otherwise, there exists an infinite set, K, such that $\lim_{\substack{k\to\infty \\ k\in K}} \Delta_k = 0$, which implies the solution of (6) corresponding to trust region radius $2\Delta_k$ can not be accepted as the trial step, in other words, the corresponding value satisfying $\rho_k(\Delta_k) \geq \eta$ but $\rho_k(2\Delta_k) < \eta$. Similar to Lemma 4, we can obtain $\rho_k(2\Delta_k) \geq \eta$ this is a contradiction.

Now, choose an infinite set $K_1 \subset \{l(k) - 1 : k = 1, 2, \cdots\}$, then from Lemma 5 we have

$$\begin{aligned} f_{l(k)} - f_{k+1} &\geq f_{r(k)} - f_{k+1} \\ &\geq \frac{1}{2}\eta \sum_{j=r(k)}^{k} \|g_j^M\| \min\left\{\Delta_j, \frac{\|g_j^M\|}{\|H_j^M\|}\right\} \\ &\geq \frac{1}{2}\eta \sum_{\substack{j=r(k)\\ k\in K_1}}^{k} \|g_j^M\| \min\left\{\Delta_j, \frac{\|g_j^M\|}{\|H_j^M\|}\right\} \\ &\geq \frac{1}{2}\eta \sum_{\substack{j=r(k)\\ k\in K_1}}^{k} \varepsilon \min\left\{\widehat{\Delta}, \frac{\varepsilon(1-\overline{\mu})}{b_2}\right\}. \end{aligned}$$

Since $k+1 \in \{l(k) - 1 : k = 1, 2, \cdots\}$ and $\{f_{l(k)}\}$ convergent, both the left side and the right side converge to zero and this is a contradiction. The proof is completed. □

We now prove the strong convergence of the algorithm.

Theorem 2 *Under Assumptions* **A1–A3***, we have*

$$\lim_{k\to\infty} \|g_k^M\| = 0.$$

Proof We prove this conclusion by contradiction. Assume that $\limsup_{k\to\infty}\|g_k^M\| = \varepsilon_1 > 0$, then there exists an infinite set, K_2, such that $\|g_k^M\| \geq \frac{\varepsilon_1}{2}$ for all $k \in K_2$.

We consider two cases:

Case 1 $\liminf_{\substack{k\to\infty\\ k\in K_2}} \Delta_k = 0$, then there exists an infinite set $K_3 \subset K_2$, such that $\lim_{\substack{k\to\infty\\ k\in K_3}} \Delta_k = 0$. Hence for $k \in K_3$ large enough, we have

$$\rho_k(\Delta_k) \geq \eta, \quad \rho_k(2\Delta_k) < \eta.$$

But similar to Lemma 4, we can obtain $\rho_k(2\Delta_k) \geq \eta$ and this is a contradiction.

Case 2

$$\liminf_{\substack{k\to\infty\\ k\in K_2}} \Delta_k = \Delta' > 0.$$

Similarly to Theorem 2 in [7], we have

$$
\begin{aligned}
f_0 - f_{k+1} &\geq \eta \sum_{i=0}^{k} Pred_i \\
&\geq \eta \sum_{\substack{i=0,\\ k\in K_2}}^{k} Pred_i \\
&\geq \eta \sum_{\substack{i=0\\ k\in K_2}}^{k} \frac{1}{2}\eta\varepsilon \min\{\Delta', \frac{\varepsilon(1-\overline{\mu})}{b_2}\}
\end{aligned}
$$

and this contradicts the fact of $f(x)$ is bounded below on Ω. This completes the proof. □

In Algorithm 2.1, if we modify H_k^M such that for all $d \in R^n$ and for all k, there exists $\delta > 0$,

$$\langle d, H_k^M d\rangle \geq \delta \|d\|^2. \tag{24}$$

Then we have

Theorem 3 *If Assumption* **A1–A3** *and* (24) *holds then we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{25}$$

Proof According to the definition μ_k and Theorem 2, Lemma 3, we only need to prove that $\lim_{k\to\infty} \|d_k\| = 0$. Assume that there exists an infinite set, K_4, and some $\varepsilon_2 > 0$ such that $\|d_k\| \geq \varepsilon_2$ for all $k \in K_4$. Since d_k is the solution of problem (6), there exists $\lambda_k \geq 0$ such that

$$(H_k^M + \lambda_k I)d_k = -g_k^M.$$

Thus we have

$$-\langle g_k^M, d\rangle - \frac{1}{2}\langle d, H_k^M d\rangle = \frac{1}{2}\langle d_k, H_k^M d_k\rangle + \lambda_k\|d_k\|^2 \geq \frac{1}{2}\delta\|d_k\|^2.$$

On the other hand, similarly to Theorem 2, we have for $k \in K_4$ large enough

$$
\begin{aligned}
f_0 - f_{k+1} &\geq \eta \sum_{i=0}^{k} Pred_i \\
&\geq \eta \sum_{\substack{i=0,\\ k\in K_4}}^{k} Pred_i \\
&\geq \frac{1}{2}\eta \sum_{\substack{i=0\\ k\in K_4}}^{k} \eta\delta\varepsilon_2
\end{aligned}
$$

Let $k \to \infty$; we have $f_0 - f_{k+1} < \infty$, but $\frac{1}{2}\eta \sum_{\substack{i=0 \\ k \in K_4}}^{k} \eta\delta\varepsilon_2 \to \infty$, this is a contradiction.

Thus $\lim_{k\to\infty} \|d_k\| = 0$ and therefore (25) holds. This completes the proof. □

4 Numerical Tests

In order to test the efficiency of our method, we conducted numerical experiments on some degenerate problems and some classic test problems. The algorithms were coded in Matlab 7.0 and run on a personal computer with a 2.93GHZ CPU processor. We chose the following five problems from CUTE collection established by Bongartz et al. [12] as our test examples:

Problem 1 (Raydan 1 function): $f(x) = \Sigma_{i=1}^{n}(exp(x_i) - x_i)$.
Problem 2 (Raydan 2 function): $f(x) = \Sigma_{i=1}^{n}\frac{i}{10}(exp(x_i) - x_i)$.
Problem 3 (Diagonal 1 function): $f(x) = \Sigma_{i=1}^{n}(exp(x_i) - ix_i)$.
Problem 4 (Diagonal 5 function): $f(x) = \Sigma_{i=1}^{n}log((exp(x_i) - exp(-x_i))$.
Problem 5 (Quadratic QF1 function): $f(x) = \frac{1}{2}\Sigma_{i=1}^{n}ix_i^2 - x_n$.

The parameters used in our algorithm are set as follows: $\overline{\mu} = 0.01$, $\eta = 0.25$, $\Delta_{min} = 0.1$, $N = 5$, $H_0 = I$, $N_0 = 5$. H_k is updated by the BFGS formulae:

$$H_{k+1} = \begin{cases} H_k, & \text{if } \delta_k^T y_k \le 0, \\ H_k + \dfrac{y_k y_k^T}{y_k^T \delta_k} - \dfrac{H_k \delta_k \delta_k^T H_k}{\delta_k^T H_k \delta_k}, & \text{if } \delta_k^T y_k > 0, \end{cases}$$

Table 1 Tests results

No.	x^i	n	$I_g/I_f(MTR)$	$I_g/I_f(TR)$
1	1	10	34/66	45/88
	1	100	101/200	101/200
	i/n	10	22/42	22/42
	i/n	100	60/118	64/126
2	1	10	353/704	353/704
	1	50	419/838	419/838
	i/n	10	167/332	167/332
	i/n	50	233/474	235/478
3	1	10	60/124	60/124
	1	50	428/874	450/922
	i/n	10	97/202	95/198
	i/n	50	285/578	316/650
4	1	10	33/64	33/64
	1	100	101/200	101/200
	1	200	143/284	143/284
	i/n	10	21/40	21/40
	i/n	100	60/118	60/118
	i/n	200	83/164	83/164
5	1	10	66/140	66/140
	1	50	356/724	402/824
	i/n	10	52/112	52/112
	i/n	50	489/998	95/200

where $\delta_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. To determine when to stop the execution of the algorithm and assert convergence, we used the criterion $\|g_k\| \leq 10^{-5}$. We also stop the execution when 500 iterations were completed without achieving convergence and denoted this as failure.

To show the efficiency of our method, we compare it with the memoryless method, i.e., $\overline{\mu} \equiv 0$. The test results are given in Table 1. Here we use *No* to denote the number of the test problems, MTR denotes the memory method and TR denotes the memoryless method, I_g and I_f denote the number of gradient estimations and the function value estimations, x^i denotes the i-th element of the initial point, and n denotes the dimension of the test problem. From the Table 1, we find that our memory method can compete with the memoryless one. In general, our method is better than the memoryless one in the number of the gradient estimations and the function value estimations.

5 Conclusion

In this paper, we considered adding memory into the unconditional trust region model. The new model includes the information of the past iterations, and therefore ensure the algorithm's behavior is dominated by the more global nature of the objective function. Global convergence is established by using non monotonic technique and numerical tests are also given to show the efficiency of the proposed algorithm.

References

1. Powell, M.J.D.: In: Mangasarian, O.L.M., Meyer, R.R., Robinson, S.M. (eds.) Convergence properties of a class minimization algorithms in nonlinear programming, vol. 2, pp. 47–67 (1975)
2. Powell, M.J.D.: On global convergence of trust region algorithms for unconstrained optimization. Math. Program. **29**, 297–303 (1984)
3. Yuan, Y.X.: Nonlinear Optimization: Trust Region Algorithms. Research Report, Institute of Computational Mathematics, Chinese Academy of Sciences (1999)
4. Sorensen, D.C.: Trust region methods for unconstrained optimization. In: Powell, M.J.D. (ed.) Nonlinear Optimization 1981, pp. 29–38. Academic, London (1982)
5. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust Region Methods. MSP-SIAM Series on Optimization. Philadelphia, PA (2000)
6. Gould, N.I.M., Lucidi, S., Roma, M., Toint, P.L.: A line search algorithm with memory for unconstrained optimization. In: De leone, R., Mauli, A., Pardalos, P., Toraldo, G. (eds.) High Performance Algorithms and Software in Nonlinear Optimization, pp. 207–223. Kluwer Academic Publishers (1998)
7. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
8. Deng, N.Y., Xiao, Y., Zhou, F.: Nonmonotone trust region algorithm. J. Optim. Theory. Appl. **76**, 259–285 (1993)
9. Ke, X.W., Han, J.Y.: A class of nonmonotone trust region algorithm for unconstrained optimization. Sci. China **28**, 488–492 (1998)
10. Toint, P.L.: A nonmonotone trust region algorithm for nonlinear programming subject to convex constraints. Math. Program. **77**, 69–94 (1997)
11. Chen, Z.W., Han, J.Y., Xu, D.C.: A nonmonotone trust region algorithm for nonlinear programming with simple bound constraints. Appl. Math. Optim. **43**, 63–85 (2000)
12. Bongartz, I., Conn, A.R., Gould, N.I., Toint, P.L.: CUTE: constrained and unconstrained testing environments. ACM T Math. Software. **21**, 123–160 (1995)