Journal of Computational Mathematics, Vol.21, No.2, 2003, 221–228.

A NEW FAMILY OF TRUST REGION ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION[*1)]

Yuhong Da Dachuan Xu
(*State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, China*)

Abstract

Trust region (TR) algorithms are a class of recently developed algorithms for nonlinear optimization. A new family of TR algorithms for unconstrained optimization, which is the extension of the usual TR method, is presented in this paper. When the objective function is bounded below and continuously differentiable, and the norm of the Hesse approximations increases at most linearly with the iteration number, we prove the global convergence of the algorithms. Limited numerical results are reported, which indicate that our new TR algorithm is competitive.

Key words: trust region method, global convergence, quasi-Newton method, unconstrained optimization, nonlinear programming.

1. Introduction

In this paper we consider the unconstrained optimization problem

$$\min f(x), \qquad x \in R^n, \tag{1.1}$$

where f is a continuous differentiable mapping from R^n to R^1. Many trust region (TR) algorithms for problem (1.1) apply the following iterative method (for instance, see [10]). At the beginning of the k-th iteration one has an estimation x_k of the required vector of variables, an $n \times n$ symmetric matrix B_k which need not be positive definite, and a trust region radius Δ_k. A TR algorithm calculates a trial step s_k by solving the "trust region subproblem":

$$\min_{d \in R^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{1.2}$$

$$\text{s. t.} \quad ||d||_2 \le \Delta_k \tag{1.3}$$

where $g_k = \nabla f(x_k)$ and B_k is an approximation to the Hessian of $f(x)$. The algorithm then computes the ratio r_k between the actual reduction and the predicted reduction in the objective function

$$r_k = \frac{\text{Ared}_k}{\text{Pred}_k} = \frac{f(x_k) - f(x_k + s_k)}{\phi_k(0) - \phi_k(s_k)}, \tag{1.4}$$

and decides whether the trial step s_k is accepted and how the next trust radius Δ_{k+1} is chosen according to the value of r_k.

---

[*] Received March 2, 2000.
[1)] Research partly supported by Chinese NSF grants 19731001 and 19801033. The second author gratefully acknowledges the support of Natoinal 973 Information Fechnology and High-Performance Software Program of China with grant No. G1998030401 and K. C. Wong Educatoin Foundation, Hong Kong.

Recently, many authors ([1-5]) give some nonmonotone trust region methods for unconstrained optimization. Toint [8] points out that the nonmonotone technique is helpful to overcome the case that the sequence of iterates follows the bottom of curved narrow valleys (a common occurrence in difficult nonlinear problems). The nonmonotone trust region algorithm presented in [2] adjusts the next trust radius Δ_{k+1} according to

$$\tilde{r}_k = \frac{f_{l(k)} - f(x_k + s_k)}{\phi_k(0) - \phi_k(s_k)}, \tag{1.5}$$

where $f_{l(k)} = \max_{0 \le j \le m(k)} \{f(k-j)\}$, $m(k) = \min\{m(k-1)+1, 2M, M_k\}$, $m(0) := 0$, $M \ge 0$ is an integer, M_k is relevant with k and is given in the specific algorithm. As pointed out in [5], however, one disadvantage of using (1.5) is that, it uses the function value at $x_{l(k)}$, which may be far away from the current point x_k.

If the matrix B_k is exactly the Hessian H_k of the objective function at x_k, and if the trust region subproblem (1.2)-(1.3) are solved exactly, it would be reasonable to use the current ratio r_k to adjust the next trust radius Δ_{k+1}. However, in practical computations, the matrix B_k is often obtained approximately (a common way is to update B_{k-1} using the pair (s_{k-1}, y_{k-1})), and the subproblem (1.2)-(1.3) are solved roughly. In such a case, it may be more reasonable to adjust the next trust radius Δ_{k+1} according to not only r_k, but the previous ratios $\{r_{k-m}, \ldots, r_k\}$, where m is some positive integer.

Following this line, we define the following quantity

$$\bar{r}_k = \sum_{i=1}^{\min\{k,m\}} w_{ki} r_{k-i+1}, \tag{1.6}$$

where $w_{ki} \in [0,1]$ is the weight of r_{k-i+1}, satisfying

$$\sum_{i=1}^{m} w_{ki} = 1. \tag{1.7}$$

In the next section, we will describe a new family of TR algorithm in which the adjusting of the next trust radius Δ_{k+1} depends on the quantity $\bar{r}_k$ in (1.6). In Section 3, we will prove the global convergence of our new TR algorithm under very mild assumptions. The numerical results, which are reported in Section 4, show that our new TR algorithm outperforms the usual TR method for the giving test problems. Conclusions and some discussions are given in Section 5.

2. The Algorithm

We now describe the new TR algorithm as follows.

Algorithm 2.1

Step 1 Given $x_1 \in R^n$, $\Delta_1 > 0$, $\varepsilon \ge 0$, $B_1 \in R^{n\times n}$ *symmetric;*
$0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 < \tau_2 < 1$, $k := 1$.

Step 2 If $||g_k||_2 \le \varepsilon$ *then stop;*
Find an approximate solution of (1.2)-(1.3), s_k.

Step 3 Choose $w_{ki} \in [0,1]$ *satisfying (1.7) and compute* r_k *and* $\bar{r}_k$ *by (1.4) and (1.6); Calculate* x_{k+1} *as follows:*

$$x_{k+1} = \begin{cases} x_k & if\ r_k \le 0, \\ x_k + s_k & otherwise; \end{cases} \tag{2.1}$$

Choose Δ_{k+1} *that satisfies*

$$\Delta_{k+1} \in \begin{cases} [\tau_3\Delta_k,\ \tau_4\Delta_k] & if\ \bar{r}_k < \tau_2, \\ [\Delta_k,\ \tau_1\Delta_k] & otherwise. \end{cases} \tag{2.2}$$

Step 4 Update B_{k+1}*;*

$k := k+1$*; go to Step 2.*

The constants $\tau_i (i = 1, \cdots, 4)$ are chosen by users. When the weight of the present ratio r_k is 1 and the other weights are all zero, the algorithm reduces to the usual TR algorithm.

3. Convergence of TR Algorithm 2.1

In this section we prove the global convergence of Algorithm 1.1 under mild assumptions. For this purpose, we assume that the predicted reduction satisfies the following relation

$$\phi_k(0) - \phi_k(s_k) \geq c_1 ||g_k||_2 \min[\Delta_k,\ \frac{||g_k||_2}{||B_k||_2}], \tag{3.1}$$

where c_1 is some positive constant. A trial step s_k satisfying relation (3.1) is normally called a "sufficient reduction" step. In real computations, it is much easier to compute a sufficient reduction step than to find the exact solution of the subproblem (1.2)-(1.3). Several techniques can be used to compute a sufficient reduction step s_k, for example the dog-leg type techniques or the way of searching in the two dimensional space spanned by the steepest descent direction and the Newton's step.

Lemma 3.1. *Assume that* $f(x)$ *is differentiable and* $\nabla f(x)$ *is uniformly Lipschitz continuous. Let* x_k *be generated by Algorithm 2.1 with* s_k *satisfying (3.1). If there exists a positive constant* δ *such that*

$$||g_k||_2 \geq \delta, \qquad for\ all\ k, \tag{3.2}$$

then there exists a constant $\tau > 0$ *such that*

$$\Delta_k \geq \frac{\tau}{M_k} \tag{3.3}$$

holds for all k*, where* M_k *is defined by*

$$M_k = 1 + \max_{1\leq i\leq k} ||B_i||_2. \tag{3.4}$$

Proof. Because $\nabla f(x)$ is uniformly continuous, there is a positive number η such that the bound

$$s_k^T[\nabla f(x+d) - \nabla f(x)] < \frac{1}{2}c_1(1-\tau_2)\delta||s_k||_2, \quad x \in R^n, \tag{3.5}$$

is satisfied for all $||d||_2 < \eta$. We show by induction that (3.3) holds with

$$\tau = \min[\Delta_1 M_1, \tau_3^m \eta M_1, \tau_3^m \delta, c_1(1-\tau_2)\tau_3^m \delta]. \tag{3.6}$$

When $k = 1$, we clearly have that $\Delta_1 \geq \frac{\tau}{M_1}$. So (3.3) holds for $k = 1$. Assume that (3.3) is true for k. Since the sequence $\{M_k; k = 1, 2, 3, \cdots\}$ increases monotonically, to prove the truth of (3.3) for $k+1$, it suffices to establish the relation

$$\Delta_{k+1} \geq \frac{\tau}{M_k}. \tag{3.7}$$

Relation (3.7) is trivial if $\Delta_{k+1} \geq \Delta_k$. Therefore we assume in the remainder of proof that Δ_{k+1} is in the range

$$\tau_3 \Delta_k \leq \Delta_{k+1} \leq \tau_4 \Delta_k. \tag{3.8}$$

In this case the technique for adjusting the radius of the trust region implies that $\bar{r}_k < \tau_2$. From the definition of $\bar{r}_k$, we can deduce that there exists an integer $\bar{k} \in [k - \min\{k, m\} + 1, k]$ such that $r_{\bar{k}} < \tau_2$:

$$f(x_{\bar{k}} + s_{\bar{k}}) - f(x_{\bar{k}}) > \tau_2 \{\phi_{\bar{k}}(x_{\bar{k}} + s_{\bar{k}}) - \phi_{\bar{k}}(0)\}. \tag{3.9}$$

If $||s_{\bar{k}}||_2 \geq \eta$, we have the bound

$$\begin{aligned} \Delta_{k+1} &\geq \tau_3^{k+1-\bar{k}} \Delta_{\bar{k}} \geq \tau_3^m \Delta_{\bar{k}} \\ &\geq \tau_3^m ||s_{\bar{k}}||_2 \geq \tau_3^m \eta \\ &\geq \tau_3^m \eta \frac{M_1}{M_k} \geq \frac{\tau}{M_k}. \end{aligned} \tag{3.10}$$

Therefore in the rest of our proof we also assume that $||s_{\bar{k}}||_2 < \eta$.

It follows from the mean vaule theorem and (3.5) that

$$\begin{aligned} f(x_{\bar{k}} + s_{\bar{k}}) - f(x_{\bar{k}}) &= \int_0^1 s_{\bar{k}}^T \nabla f(x_{\bar{k}} + \theta s_{\bar{k}}) \, d\theta \\ &= s_{\bar{k}}^T g_{\bar{k}} + \int_0^1 s_{\bar{k}}^T \{\nabla f(x_{\bar{k}} + \theta s_{\bar{k}}) - g_{\bar{k}}\} \, d\theta \\ &< s_{\bar{k}}^T g_{\bar{k}} + \frac{1}{2} c_1 (1 - \tau_2) \delta ||s_{\bar{k}}||_2, \end{aligned} \tag{3.11}$$

which with (3.9) gives the bound

$$(1 - \tau_2)\{s_{\bar{k}}^T g_{\bar{k}} + \frac{1}{2} c_1 \delta ||s_{\bar{k}}||_2\} > \frac{1}{2} \tau_2 s_{\bar{k}}^T B_{\bar{k}} s_{\bar{k}}. \tag{3.12}$$

Moreover, we have by (3.1) and (3.2) that

$$-s_{\bar{k}}^T g_{\bar{k}} - \frac{1}{2} s_{\bar{k}}^T B_{\bar{k}} s_{\bar{k}} \geq c_1 \delta \min[\Delta_{\bar{k}}, \ \frac{\delta}{||B_{\bar{k}}||_2}]. \tag{3.13}$$

By adding $(1 - \tau_2)$ times this inequality to (3.12), we can get that

$$\begin{aligned} \Delta_{\bar{k}}^2 ||B_k||_2 &\geq ||s_{\bar{k}}||_2^2 ||B_{\bar{k}}||_2 \\ &\geq -s_{\bar{k}}^T B_{\bar{k}} s_{\bar{k}} \\ &> 2(1 - \tau_2) c_1 \delta \min[\Delta_{\bar{k}}, \ \frac{\delta}{||B_{\bar{k}}||_2}] - (1 - \tau_2) c_1 \delta ||s_{\bar{k}}||_2 \\ &\geq c_1 (1 - \tau_2) \delta \{2 \min[\Delta_{\bar{k}}, \ \frac{\delta}{||B_{\bar{k}}||_2}] - \Delta_{\bar{k}}\} \\ &= c_1 (1 - \tau_2) \delta \min[\Delta_{\bar{k}}, \ 2\frac{\delta}{||B_{\bar{k}}||_2} - \Delta_{\bar{k}}]. \end{aligned} \tag{3.14}$$

Using the above condition, we can give a constant lower bound on the product $\Delta_{\bar{k}} ||B_{\bar{k}}||_2$. In fact, if $\Delta_{\bar{k}} ||B_{\bar{k}}||_2 \leq \delta$, it follows from (3.14) that

$$\Delta_{\bar{k}} ||B_{\bar{k}}||_2 > c_1 (1 - \tau_2) \delta, \qquad \Delta_{\bar{k}} > \frac{\delta}{||B_{\bar{k}}||_2}. \tag{3.15}$$

Thus the following relation always holds

$$\Delta_{\bar{k}}||B_{\bar{k}}||_2 > \min[c_1(1-\tau_2)\delta,\ \delta] \geq \frac{\tau}{\tau_3^m}. \tag{3.16}$$

By (3.8), (3.16), (3.4) and the definition of τ, we can then obtain

$$\begin{aligned} \Delta_{k+1} &\geq \tau_3\Delta_k \geq \tau_3^{k+1-\bar{k}}\Delta_{\bar{k}} \\ &\geq \tau_3^m\Delta_{\bar{k}} > \frac{\tau}{||B_{\bar{k}}||_2} \\ &\geq \frac{\tau}{M_{\bar{k}}} \geq \frac{\tau}{M_k}. \end{aligned} \tag{3.17}$$

Thus (3.3) also holds for $k+1$. Therefore by induction, (3.3) holds for all $k \geq 1$. □

Powell [7] showed the global convergence for the usual TR method under the assumption that

$$||B_k||_2 \leq c_2 + c_3 k, \quad k = 1, 2, 3, \cdots, \tag{3.18}$$

where c_2 and c_3 are constants. To prove the global convergence of Algorithm 2.1 under (3.18). we draw the following lemma from [7] or [9].

Lemma 3.2. *Let $\{\Delta_k\}$ and $\{M_k\}$ be two sequences such that $\Delta_k \geq \dfrac{\tau}{M_k} \geq 0$ for all k, where τ is a positive constant. Let J be a subset of $\{1, 2, 3, \cdots\}$. Assume that*

$$\Delta_{k+1} \leq \tau_1\Delta_k, \quad k \in J \tag{3.19}$$

$$\Delta_{k+1} \leq \tau_4\Delta_k, \quad k \notin J \tag{3.20}$$

$$M_{k+1} \geq M_k, \quad k \geq 1 \tag{3.21}$$

$$\sum_{k\in J}\frac{1}{M_k} < \infty, \tag{3.22}$$

where $\tau_1 > 1$, $\tau_4 < 1$ are positive constants. Then

$$\sum_{k=1}^{\infty}\frac{1}{M_k} < \infty. \tag{3.23}$$

Lemma 3.3. *Assume that the conditions of Lemma 3.1 holds. If $\{f(x_k)\}$ is bounded below, we have that (3.23) holds.*

Proof. Define the set $J = \{k|\bar{r}_k \geq \tau_2\}$. Then (3.19) and (3.20) follow the update formula (2.2). Further, for each index $k \in J$, we define $S_k = \{\tilde{k} : r_{\tilde{k}} \geq \tau_2, \tilde{k} \in [k - \min\{k, m\} + 1, k]\}$. If $\bar{r}_k \in J$, we know from the definition of $\bar{r}_k$ that there exists an integer $\tilde{k} \in [k - \min\{k, m\} + 1, k]$ such that $r_{\tilde{k}} \geq \tau_2$. Thus S_k is nonempty, which with the nonmonotonicity of M_k implies that

$$\sum_{\tilde{k}\in S_k}\frac{1}{M_{\tilde{k}}} \geq \frac{1}{M_k}, \qquad \text{for all } k \in J. \tag{3.24}$$

Since $\{f(x_k)\}$ is bounded below, we have from the definitions of J and S_k, Lemma 3.1 and

(3.24) that

$$\begin{aligned} +\infty &> \sum_{k=1}^{\infty}(f(x_k)-f(x_{k+1})) \\ &\geq \frac{1}{m}\sum_{k\in J}\sum_{\tilde{k}\in\tilde{S}_k}(f(x_{\tilde{k}})-f(x_{\tilde{k}+1})) \\ &\geq \frac{\tau_2}{m}\sum_{k\in J}\sum_{\tilde{k}\in\tilde{S}_k}[\phi_{\tilde{k}}(0)-\phi_{\tilde{k}}(s_{\tilde{k}})] \\ &\geq \frac{c_1\tau_2\delta}{m}\sum_{k\in J}\sum_{\tilde{k}\in\tilde{S}_k}\min[\Delta_{\tilde{k}},\frac{\delta}{||B_{\tilde{k}}||_2}] \\ &\geq \frac{c_1\tau_2\delta\min[\tau,\delta]}{m} \end{aligned} \tag{3.25}$$

Therefore by Lemma 3.2, we know that this lemma is true. □

Now we are ready to give our main convergence theorem.

Theorem 3.4. *Assume that $f(x)$ is differentiable and $\nabla f(x)$ is uniformly Lipschitz continuous. Let x_k be generated by Algorithm 2.1 with s_k satisfying (3.1). If (3.18) holds, if $\varepsilon = 0$ in Algorithm 2.1, and if $\{f(x_k)\}$ is bounded below, then either $g_k = 0$ for some k or the following relation holds*

$$\liminf_{k\to\infty}||g_k||_2 = 0. \tag{3.26}$$

Proof. We prove by contradiction. Assume that the theorem is not true, namely, condition (3.2) holds for some constant $\delta > 0$. Then we have by Lemma 3.3 that (3.23) holds. However, it follows from (3.18) that

$$\sum_{k=1}^{\infty}\frac{1}{M_k} = \infty, \tag{3.27}$$

which contradicts (3.23). The contradiction shows that the theorem is true. □

Remark 3.1. *It may be not possible to strengthen Theorem 3.4 about the new TR algorithm. This is because, if $\{M_k; k = 1, 2, 3, \cdots\}$ is any nondecreasing sequence of positive numbers such that condition (3.23) holds, we can use the counter-example presented by Powell [7] to show the judgement.*

Remark 3.2. *In practical computations, for $k \geq 2$ one may recursively define $\bar{r}_k$ as follows:*

$$\bar{r}_k = \mu_k r_k + (1-\mu_k)\bar{r}_{k-1}, \tag{3.28}$$

where $\mu \in (0, 1)$ is some constant. In this case, we can establish the global convergence result for the choice (3.28) in a similar way.

4. Numerical Experiments

We tested the usual TR method (UTR) and the new TR method (NTR) with double precisions on an SGI Indigo workstation. The codes are edited by FORTRAN language and are based on Y. Yuan's ones for the usual TR method. The test problems and the used initial points were taken from Moré, Garbow and Hillstrom [6]. In practical computations, we prefer to the following choice:

$$\bar{r}_k = w_1 r_k + (1-w_1)\bar{r}_{k-1}, \tag{4.1}$$

where $\bar{r}_1 = r_1$. The above choice of $\bar{r}_k$ is easier to calculate. If $w_1 = 1.0$, then Algorithm 2.1 with $\bar{r}_k$ given by (4.1) is corresponding to the usual TR method. For each problem, the stopping condition is

$$\|g_k\| \le 10^{-6}. \tag{4.2}$$

Algorithm 2.1 with $\bar{r}_k$ given by (4.1) is tested with different values of w_1 in $(0,1]$. We find that the choice $w_1 = 0.9$ gives the best numerical results for the given test problems. Here we only list the numerical results of $w_1 = 0.9$ and 1.0, that are rather typical. See Table 4.1, where n, n_f and n_g mean the number of iterations, the number of function evaluations, and the number of gradient evaluations, respectively. In the table, the unconstrained optimization problems are numbered in the same way as in [6]. For example, "MGH2" means problem 2 in [6]. From Table 4.1, we see that the new TR method performs better than the usual TR method for four of the test problems, whereas for the other two problems, the usual TR method requires fewer function evaluations and gradient evaluations. On the whole, the new TR method with $w_1 = 0.9$ performs better than the usual TR method.

Table 4.1 Numerical comparisons of UTR and NTR

problems	UTR	NTR($w_1 = 0.9$)
MGH2	43/44/40	39/40/38
MGH4	208/209/172	99/100/78
MGH8	100/101/88	87/88/80
MGH9	11/12/10	10/11/9
MGH10	38/39/38	35/36/36
MGH15	59/60/52	83/84/76
MGH17	55/56/49	45/46/40
MGH18	32/33/26	59/60/47

5. Conclusions

A new family of TR algorithms for unconstrained optimization has been presented. Under mild assumptions, the new family of TR algorithms are proved to be globally convergent. Preliminary numerical results have been reported, which showed that the new TR method may be competitive with the usual TR method.

Although Theorem 3.4 allows a large range of the weights of $\bar{r}_k$, it is worth studying how to choose the optimal values for them in practical computations.

Acknowledgments. The authors would like to thank Professor Yaxiang Yuan, who provided us with FORTRAN codes for the usual TR algorithm. Thanks are also due to the two anonymous referees for their useful comments on the paper.

References

[1] N.Y. Deng, Y. Xiao and F.J. Zhou, A nonmonotonic trust region algorithm, *Journal of Optimization Theory and Applications*, **76**(1993), 259-285.

[2] Ke Xiaowu and Han Jiye, Global convergence of a class of new trust region algorithms (in Chinese), *ACTA Mathematicae Applicatae Sinica*, **18**(1995), 608-615.

[3] Ke Xiaowu and Han Jiye, A class of nonmonotone trust region algorithms for unconstrained optimization (in Chinese), *Science in China (Series A)*, **28**(1998), 488-492.

[4] Ke Xiaowu, Liu Guanghui and Xu Dachuan, A nonmonotone trust region algorithm for unconstrained optimization, *Chinese Science Bulletin*, **41**(1996), 197-201.

[5] Li Zhengfeng and Deng Naiyang, A new family of nonmonotonic trust-region algorithms and its properties (in Chinese), *ACTA Mathematicae Applicatae Sinica*, **22**(1999), 457-465.

[6] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, **7**(1981), 17-41.

[7] M.J.D. Powell, On the global convergence of trust region algorithms for unconstrained minimization, *Mathematical Programming*, **29**(1984), 297-303.

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

[9] Y. Yuan, Nonlinear optimization: trust region algorithms, Research Report, Institute of Computational Mathematics, Chinese Academy of Sciences, 1999.

[10] Y. Yuan, Convergence properties of trust region method (in Chinese), *Mathematica Numerica Sinica*, **16**(1994), 333-346.