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

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

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 $f_{l(k)} = \max\limits_{0 \le j \le m(k)} \{f(k-j)\}$, $m(k) = \min\{m(k-1)+1, 2M, M_k\}$, $m(0) := 0$, $M \ge 0$ 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:*

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