Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 3, No. 3, pp. 539–554, September 2007

SOME EXTENSIONS OF THE TRUST REGION METHOD*

ZHEN-JUN SHI, JIE SHEN AND JINHUA GUO

Abstract: In this paper, we develop some new properties of the trust region method for unconstrained optimization problems by generalizing Cauchy point to a general form. These new extensions enable us to simplify the subproblems and design some new and effective trust region methods. Moreover, we propose several implementable trust region algorithms in which the subproblem is simple and easily solvable. Preliminary numerical results show that some new trust region algorithms are available and efficient in practical computation.

Key words: *unconstrained optimization, trust region method, global convergence*

Mathematics Subject Classification: *90C30, 65K05, 49M37*

1 Introduction

The trust region method is an important technique for solving optimization problems, due to its strong convergence and robustness (e.g. [4, 5, 7, 8, 9, 12, 13, 16, 18, 27]). The trust region method for unconstrained optimization problems defines each iterate as the approximate minimizer of a relatively simple model function within a region in which the algorithm trusts that the model function behaves like $f(x)$ at the current iterate (e.g. [3, 6, 15, 17, 19, 25, 26]). Unlike line search methods in which the search direction is to be chosen firstly at each iteration, the trust region method not only avoids the line search procedure, but also produces the new iterates by solving some subproblems and has strong global convergence (e.g. [14, 15, 20]). In trust region methods, the direction and step size are chosen simultaneously. In general, the direction changes whenever the size of the trust region is altered ([2, 23, 24, 28]).

The advantages of the trust region method are strong global convergence and robustness. In order to analyze the convergence, one often uses the Cauchy point to obtain some useful convergence properties ([3, 10, 12]). Can we generalize the Cauchy point to a general form to obtain some new convergence properties? The answer is yes.

In this paper, we develop some new properties of the trust region method for unconstrained optimization problems by generalizing the Cauchy point to a general form. These new extensions enable us to design some new and effective trust region methods. We propose several simple and implementable trust region algorithms in which the subproblem is easy to solve. Preliminary numerical results show that the new trust region algorithms are available and efficient in practical computation.

*The work was supported in part by NSF DMI DMI-0514900 and NSF CNS-0521142, USA.

The rest of this paper is organized as follows. In the next section, we give some preliminary results on trust region methods. In Section 3, we develop some new properties of the trust region method. In Section 4, we propose several simple and implementable trust region methods. Some conclusions are summarized in Section 5.

2 Trust Region Method

Trust-region methods produce a trial step by minimizing a quadratic model of the objective function subject to a ball constraint. Because of this restriction, trust-region methods are sometimes known as restricted-step methods. In this section, we summarize some properties of trust-region methods. For an in-depth overview of trust-region methods, see Conn, Gould, and Toint's book ([3]) and Nocedal and Wright's book [12].

Consider an unconstrained optimization problem

$$\min_{x\in R^n} f(x), \tag{2.1}$$

where R^n is an n-dimensional Euclidean space and $f : R^n \to R^1$ is a continuously differentiable function. Denote $g(x) = \nabla f(x), G(x) = \nabla^2 f(x)$. If x_k (k=0,1,2,...) is the current iterate, then we denote $f_k = f(x_k), g_k = \nabla f(x_k)$ and $G_k = \nabla^2 f(x_k)$. Suppose x^* is a solution or a stationary point of the unconstrained optimization problem, we denote $f^* = f(x^*), g^* = \nabla f(x^*)$ and $G^* = \nabla^2 f(x^*)$.

In trust region method, we need to seek a solution to the subproblem

$$\min_{p\in R^n} m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p, \quad s.t.\ \|p\| \le \Delta_k, \tag{2.2}$$

where Δ_k is a trust region radius and B_k is an approximation to G_k. We define $\|\cdot\|$ to be the Euclidean norm, and let the solution p_k^* to (2.2) be a minimizer of $m_k(p)$ in the ball with the radius Δ_k. Thus, the trust region method requires us to solve a sequence of subproblems (2.2) in which the objective function and constraint (which can be written as $p^T p \le \Delta_k^2$) are both quadratic.

The first issue to arise in defining a trust region method is the strategy for choosing the trust region radius Δ_k at each iteration. We should make a choice on the agreement between the model m_k and the objective function f at the previous iterate. Given a step p_k we define the ratio

$$\rho_k = \frac{f_k - f(x_k + p_k)}{m_k(0) - m_k(p_k)}, \tag{2.3}$$

where the numerator and the denominator are respectively called actual reduction and predicted reduction. Note that since the step p_k is obtained by minimizing the model m_k over a region that includes the step $p = 0$, the predicted reduction will always be nonnegative. Thus if ρ_k is negative, the new objective value $f(x_k + p_k)$ is greater than the current value f_k, so the step must be rejected.

On the other hand, if ρ_k is close to 1, there is good agreement between the model m_k and the function f over this step, so it is safe to expand the trust region for the next iteration. If ρ_k is positive but not close to 1, we do not alter the trust region. But if it is close to zero or negative, we shrink the trust region. The following algorithm describes the process [12].

Algorithm 2.1 (Trust Region Algorithm).

Given $\overline{\Delta} > 0$, $\Delta_0 \in (0, \overline{\Delta})$, and $\eta \in [0, \frac{1}{4})$;

For $k = 0, 1, 2, ...$
    Obtain p_k by (or approximately) solving (2.2);
    Evaluate ρ_k from (2.3);
    if $\rho_k < \frac{1}{4}$ then
        $\Delta_{k+1} = \frac{1}{4}\|p_k\|$
    else
        if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$ then
            $\Delta_{k+1} = \min(2\Delta_k, \overline{\Delta})$
        else
            $\Delta_{k+1} = \Delta_k$;
    if $\rho_k > \eta$ then
        $x_{k+1} = x_k + p_k$
    else
        $x_{k+1} = x_k$;
end(for).

To turn Algorithm 2.1 into a practical algorithm, we need to focus on solving (2.2). We expect some approximate solutions of (2.2) to achieve at least as much reduction in m_k as the reduction achieved by the so-called Cauchy point. This point is simply a minimizer of m_k along the steepest descent direction $-g_k$, subject to the trust region bound. It is enough to find an approximate solution p_k to (2.2) that lies within the trust region and get a sufficient reduction in the model. The sufficient reduction can be quantified in terms of the Cauchy point, which we denote by p_k^c and define in terms of the following simple procedure.

Algorithm 2.2 (Cauchy Point Calculation).

Find the vector p_k^s that solves a linear version of (2.2), i.e.,

$$p_k^s = \arg\min_{p \in R^n} (f_k + g_k^T p), \quad s.t. \ \|p\| \leq \Delta_k; \tag{2.4}$$

Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^s)$ subject to $\|\tau p_k^s\| \leq \Delta_k$ and set $p_k^c = \tau_k p_k^s$.

In fact,

$$p_k^s = -\frac{\Delta_k}{\|g_k\|} g_k,$$

and

$$p_k^c = -\tau_k \frac{\Delta_k}{\|g_k\|} g_k,$$

where

$$\tau_k = \begin{cases} 1, & \text{if } g_k^T B_k g_k \leq 0; \\ \min(\|g_k\|^3/(\Delta_k g_k^T B_k g_k), 1), & \text{otherwise.} \end{cases} \tag{2.5}$$

If Algorithm 2.1 produces an approximate solution p_k to the subproblem (2.2) that satisfies the estimate

$$m_k(0) - m_k(p_k) \geq c_1 \|g_k\| \min\Big(\Delta_k, \frac{\|g_k\|}{\|B_k\|}\Big), \tag{2.6}$$

for some constant $c_1 \in (0, 1]$, then we can show that the Cauchy point p_k^c satisfies (2.6) with $c_1 = \frac{1}{2}$, see ([12]).

Lemma 2.3. *The Cauchy point p_k^c satisfies (2.6) with $c_1 = \frac{1}{2}$, that is*

$$m_k(0) - m_k(p_k) \geq \frac{1}{2}\|g_k\| \min\Big(\Delta_k, \frac{\|g_k\|}{\|B_k\|}\Big). \tag{2.7}$$

Theorem 2.4. *Let p_k be any vector such that $\|p_k\| \leq \Delta_k$ and $m_k(0)-m_k(p_k) \geq c_2(m_k(0)-m_k(p_k^c))$. Then p_k satisfies (2.6) with $c_1 = \frac{c_2}{2}$. In particular, if p_k is the exact solution p_k^* of (2.2), then it satisfies (2.6) with $c_1 = \frac{1}{2}$.*

For generality, we allow the length of the approximate solution p_k of (2.2) to exceed the trust region bound, i.e.,

$$\|p_k\| \leq \gamma\Delta_k, \tag{2.8}$$

where $\gamma \in [1, +\infty)$.

Theorem 2.5. *Let $\eta = 0$ in Algorithm 2.1. Suppose that $\|B_k\| \leq \beta$ for some constant β, f is continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \leq f(x_0)\}$. Then, for all approximate solutions of (2.2) satisfying the inequalities (2.6) and (2.8), we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{2.9}$$

Theorem 2.6. *Let $\eta \in (0, \frac{1}{4})$ in Algorithm 2.1. Suppose that $\|B_k\| \leq \beta$ for some constant β, f is Lipschitz continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \leq f(x_0)\}$. Then, for all approximate solutions of (2.2) satisfying the inequalities (2.6) and (2.8), we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{2.10}$$

Moré and Sorensen described a safeguarded version of the root-finding Newton method ([10]) in which the approximate solution p satisfies the conditions (2.8) and

$$m_k(0) - m_k(p) \geq c_1(m_k(0) - m_k(p_k^*)) \tag{2.11}$$

for $c_1 \in (0, 1]$ and $\gamma \geq 1$, where p_k^* is the exact solution to (2.2).

3 New Properties of Trust Region Method

In line search methods, we first choose a search direction d_k and then find a new iterate along the direction at each step. The distance to move along d_k can be found by approximately solving the one-dimensional minimization problem

$$\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k).$$

Set

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.1}$$

and complete one iteration.

The search direction d_k is generally required to satisfy

$$g_k^T d_k < 0, \tag{3.2}$$

which guarantees that d_k is a descent direction of $f(x)$ at x_k. In order to guarantee the global convergence, we sometimes require d_k to satisfy the sufficient descent condition

$$g_k^T d_k \leq -c\|g_k\|^2, \tag{3.3}$$

where $c > 0$ is a constant. Moreover, the angle property

$$\cos\langle -g_k, d_k\rangle = -\frac{g_k^T d_k}{\|g_k\| \cdot \|d_k\|} \geq \tau_0 \tag{3.4}$$

with $\tau_0 : 1 \geq \tau_0 > 0$ is commonly used in proving the global convergence.

In the trust region method, we choose a search direction d_k to satisfy (3.2), (3.3) or (3.4) and solve the following subproblem

$$\min m_k(\tau d_k), \quad s.t. \ \|\tau d_k\| \leq \Delta_k \tag{3.5}$$

to obtain a solution $p_k^l = \tau_k d_k$, where

$$\tau_k = \begin{cases} \Delta_k/\|d_k\|, & \text{if } d_k^T B_k d_k \leq 0; \\ \min(-g_k^T d_k / d_k^T B_k d_k, \Delta_k/\|d_k\|), & \text{otherwise.} \end{cases} \tag{3.6}$$

Obviously, p_k^l reduces to the Cauchy point whenever $d_k = -g_k$. We can use p_k^l to obtain some new convergence properties of trust region method. As we can see that the point p_k^l is the minimizer of m_k along the direction d_k, subject to the trust region bound. For global convergence, it is enough to find an approximate solution p_k that lies within the trust region and obtain a sufficient reduction in the model. The sufficient reduction can be quantified in terms of the point p_k^l.

If the approximate solution p_k to the subproblem (2.2) satisfies (2.8) and

$$m_k(0) - m_k(p_k) \geq -c_1 \frac{g_k^T d_k}{\|d_k\|} \min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\| \cdot \|B_k\|}\Big) \tag{3.7}$$

for $c_1 \in (0, 1]$, then we can obtain some generalized convergence theorems.

Lemma 3.1. *The point p_k^l satisfies (3.7) with $c_1 = \frac{1}{2}$, i.e.,*

$$m_k(0) - m_k(p_k^l) \geq -\frac{1}{2} \frac{g_k^T d_k}{\|d_k\|} \min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\| \cdot \|B_k\|}\Big). \tag{3.8}$$

Proof. We first consider the case of $d_k^T B_k d_k \leq 0$. Since

$$\begin{aligned} m_k(p_k^l) - m_k(0) &= m_k(\tau_k d_k) - m_k(0) \\ &= \tau_k g_k^T d_k + \frac{1}{2}\tau_k^2 d_k B_k d_k \\ &\leq \tau_k g_k^T d_k = \Delta_k \frac{g_k^T d_k}{\|d_k\|} \\ &\leq \frac{1}{2} \frac{g_k^T d_k}{\|d_k\|} \min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\| \cdot \|B_k\|}\Big), \end{aligned}$$

we obtain that (3.8) holds in this case.

For the next case of $d_k^T B_k d_k > 0$ and $-\frac{g_k^T d_k}{d_k^T B_k d_k} \leq \frac{\Delta_k}{\|d_k\|}$, we have $\tau_k = -\frac{g_k^T d_k}{d_k^T B_k d_k}$, and thus

$$\begin{aligned} m_k(p_k^l) - m_k(0) &= m_k(\tau_k d_k) - m_k(0) \\ &= \tau_k g_k^T d_k + \frac{1}{2}\tau_k^2 d_k B_k d_k \\ &= -\frac{1}{2} \frac{(g_k^T d_k)^2}{d_k^T B_k d_k} \leq -\frac{1}{2} \frac{(g_k^T d_k)^2}{\|d_k\|^2 \|B_k\|} \\ &\leq \frac{1}{2} \frac{g_k^T d_k}{\|d_k\|} \min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\| \cdot \|B_k\|}\Big). \end{aligned}$$

Therefore (3.8) also holds.

In the remaining case, we have $-\frac{g_k^T d_k}{d_k^T B_k d_k} > \frac{\Delta_k}{\|d_k\|}$. Therefore, $d_k^T B_k d_k < -\frac{\|d_k\| g_k^T d_k}{\Delta_k}$, $\tau_k = \frac{\Delta_k}{\|d_k\|}$, and consequently

$$\begin{aligned} m_k(p_k^l) - m_k(0) &= \frac{\Delta_k}{\|d_k\|} g_k^T d_k + \frac{1}{2}\Big(\frac{\Delta_k}{\|d_k\|}\Big)^2 d_k^T B_k d_k \\ &\le \frac{1}{2}\frac{g_k^T d_k}{\|d_k\|}\Delta_k \\ &\le \frac{1}{2}\frac{g_k^T d_k}{\|d_k\|}\min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\|\cdot\|B_k\|}\Big), \end{aligned}$$

yielding the desired result. The proof is finished. □

It is obvious that Lemma 2.3 is a corollary of Lemma 3.1 whenever $d_k = -g_k$.

Theorem 3.2. *Let p_k be any vector such that $\|p_k\| \le \Delta_k$ and $m_k(0) - m_k(p_k) \ge c_2(m_k(0) - m_k(p_k^l))$. Then p_k satisfies (3.7) with $c_1 = \frac{c_2}{2}$. In particular, if p_k is the exact solution p_k^* to (2.2), then it satisfies (3.7) with $c_1 = \frac{1}{2}$.*

Proof. Since $\|p_k\| \le \Delta_k$, we have from (3.8) that

$$\begin{aligned} m_k(0) - m_k(p_k) &\ge c_2(m_k(0) - m_k(p_k^l)) \\ &\ge -\frac{c_2}{2}\frac{g_k^T d_k}{\|d_k\|}\min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\|\cdot\|B_k\|}\Big), \end{aligned}$$

giving the result. □

Remark 3.3. Let $S(d_k)$ denote the set of all p satisfying (3.7), $S(-g_k)$ denote the set of all p satisfying (2.6), and c_1 in (2.6) and (3.7) is the same constant. Then

$$S(-g_k) \subseteq S(d_k).$$

In fact, since

$$-\frac{g_k^T d_k}{\|d_k\|} \le \|g_k\|,$$

by (2.6) and (3.7), we have

$$\forall p \in S_k(-g_k) \Rightarrow p \in S(d_k).$$

This shows that (3.7) has a wider scope for p at the kth iteration. Moreover, Theorem 2.4 is a corollary of Theorem 3.2 whenever $d_k = -g_k$.

Theorem 3.4. *Let $\eta = 0$ in Algorithm 2.1. Suppose that $\|B_k\| \le \beta$ for some constant β, f is continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \le f(x_0)\}$. Then, for all approximate solutions of (2.2) satisfying inequalities (2.8) and (3.7) with d_k satisfying (3.2), we have*

$$\liminf_{k\to\infty}\Big(\frac{-g_k^T d_k}{\|d_k\|}\Big) = 0. \tag{3.9}$$

Proof. We first perform some technical manipulation with the ratio ρ_k from (2.3),

$$\begin{aligned} |\rho_k - 1| &= \Big|\frac{(f_k - f(x_k+p_k)) - (m_k(0) - m_k(p_k))}{m_k(0) - m_k(p_k)}\Big| \\ &= \Big|\frac{m_k(p_k) - f(x_k+p_k)}{m_k(0) - m_k(p_k)}\Big|. \end{aligned}$$

By Taylor theorem we have

$$f(x_k+p_k) = f_k + g_k^T p_k + \int_0^1 [g(x_k+tp_k) - g_k]^T p_k dt.$$

It follows from the definition of m_k that

$$\begin{aligned} |m_k(p_k) - f(x_k+p_k)| &= \Big|\frac{1}{2}p_k^T B_k p_k - \int_0^1 [g(x_k+p_k) - g_k]^T p_k dt\Big| \\ &\le (\beta/2)\|p_k\|^2 + C_4(p_k)\|p_k\|, \end{aligned}$$

where we can make the scalar $C_4(p_k)$ arbitrarily small by restricting the size of p_k.

Suppose for contradiction that (3.9) doesn't hold. Then there exists an $\epsilon > 0$ such that

$$-\frac{g_k^T d_k}{\|d_k\|} \ge \epsilon, \ \forall k. \tag{3.10}$$

From (3.7), we have

$$m_k(0) - m_k(p_k) \ge c_1 \epsilon \min\Big(\Delta_k, \frac{\epsilon}{\beta}\Big). \tag{3.11}$$

Using (3.11) and (2.8), we have

$$|\rho_k - 1| \le \frac{\gamma\Delta_k(\beta\gamma\Delta_k/2 + C_4(p_k))}{c_1\epsilon\min(\Delta_k, \epsilon/\beta)}. \tag{3.12}$$

By choosing $\overline{\Delta}$ to be small enough and noting that $\|p_k\| \le \gamma\Delta_k \le \gamma\overline{\Delta}$, we can ensure that the term in parentheses in the numerator of (3.12) satisfies the bound

$$\beta\gamma\Delta_k/2 + C_4(p_k) < \frac{c_1\epsilon}{4\gamma}. \tag{3.13}$$

By choosing $\overline{\Delta}$ even smaller, if necessary, to ensure that $\Delta_k \le \overline{\Delta} \le \epsilon/\beta$, it follows from (3.12) that

$$|\rho_k - 1| < \frac{\gamma\Delta_k c_1\epsilon/(4\gamma)}{c_1\epsilon\Delta_k} = \frac{1}{4}.$$

Therefore, $\rho_k > \frac{3}{4}$, and by the use of Algorithm 2.1, we have $\Delta_{k+1} \ge \Delta_k$ whenever Δ_k falls below the threshold $\overline{\Delta}$. It follows that reduction of Δ_k (by a factor of $\frac{1}{4}$) can occur in the algorithm only if

$$\Delta_k \ge \overline{\Delta},$$

and therefore we conclude that

$$\Delta_k \ge \min(\Delta_K, \overline{\Delta}/4), \ \forall k \ge K \tag{3.14}$$

for sufficiently large K. Suppose that there is an infinite subsequence N such that $\rho_k \geq \frac{1}{4}$ for $k \in N$. If $k \in N$ and $k \geq K$, it follows from (3.11) that

$$\begin{aligned} f_k - f_{k+1} &= f_k - f(x_k + p_k) \\ &\geq \frac{1}{4}[m_k(0) - m_k(p_k)] \\ &\geq \frac{1}{4}c_1\epsilon \min(\Delta_k, \epsilon/\beta). \end{aligned}$$

Since f is bounded from below, it follows from this inequality that

$$\lim_{k\in N, k\to\infty} \Delta_k = 0,$$

contradicting (3.14). Hence no such infinite subsequence N can exist, and we must have $\rho_k < \frac{1}{4}$ for all sufficiently large k. In this case, Δ_k will eventually be reduced by a factor of $\frac{1}{4}$ at every iteration, and we have $\lim_{k\to\infty} \Delta_k = 0$, which again contradicts (3.14). Hence, our original assertion (3.10) must be false, resulting in (3.9). □

Corollary 3.5. *Let $\eta = 0$ in Algorithm 2.1. Suppose that $\|B_k\| \leq \beta$ for some constant β, f is continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \leq f(x_0)\}$. Then, for all approximate solutions of (2.2) satisfying inequalities (2.8) and (3.7) with d_k satisfying (3.4), we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.15}$$

It is apparent that Corollary 3.5 can be proved from Theorem 2.5.

Theorem 3.6. *Let $\eta \in (0, \frac{1}{4})$ in Algorithm 2.1. Suppose that $\|B_k\| \leq \beta$ for some constant β, f is Lipschitz continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \leq f(x_0)\}$. Then, for all approximate solutions of (2.2) satisfying inequalities (2.8) and (3.7) with d_k satisfying (3.4), we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.16}$$

Proof. Consider any index m such that $\|g_m\| \neq 0$. If we use β_1 to denote the Lipschitz constant for $g(x)$ on the level set $\{x|\ f(x) \leq f(x_0)\}$, we have

$$\|g(x) - g_m\| \leq \beta_1\|x - x_m\|,$$

for all x in the level set. Hence, by defining the scalars

$$\epsilon = \frac{\|g_m\|}{2}, \quad R = \frac{\|g_m\|}{2\beta_1} = \frac{\epsilon}{\beta_1},$$

and the ball

$$B(x_m, R) = \{x|\ \|x - x_m\| \leq R\},$$

for $x \in B(x_m, R)$, we have

$$\begin{aligned} \|g(x)\| &\geq \|g_m\| - \|g_m - g(x)\| \\ &\geq \|g_m\| - \beta_1\|x - x_m\| \\ &\geq \beta_1 R = \epsilon. \end{aligned}$$

If the entire sequence $\{x_k\}_{k\geq m}$ stays inside the ball $B(x_m, R)$, we would have $\|g_k\| \geq \epsilon > 0$ for all $k \geq m$. The reasoning in the proof of Theorem 3.4 can be used to show that this scenario does not occur. Therefore, the sequence $\{x_k\}_{k\geq m}$ eventually leaves $B(x_m, R)$.

Let the index $l \geq m$ be such that x_{l+1} is the first iterate outside $B(x_m, R)$ after x_m. Since

$$\|g_k\| \geq -\frac{g_k^T d_k}{\|d_k\|} \geq \tau_0 \|g_k\| \geq \tau_0 \epsilon$$

for $k = m, m+1, ..., l$, we can use (3.11) to write

$$\begin{aligned} f_m - f_{l+1} &= \sum_{k=m}^{l} (f_k - f_{k+1}) \\ &\geq \sum_{k=m, x_k \neq x_{k+1}}^{l} \eta [m_k(0) - m_k(p_k)] \\ &\geq \sum_{k=m, x_k \neq x_{k+1}}^{l} \eta c_1 \tau_0 \epsilon \min\Big(\Delta_k, \frac{\tau_0 \epsilon}{\beta}\Big). \end{aligned}$$

If $\Delta_k \leq \tau_0 \epsilon / \beta$ for all $k = m, m+1, ..., l$, we have

$$f_m - f_{l+1} \geq \eta c_1 \tau_0 \epsilon \sum_{k=m, x_k \neq x_{k+1}} \Delta_k \geq \eta c_1 \tau_0 \epsilon R = \eta c_1 \tau_0 \epsilon^2 / \beta_1. \tag{3.17}$$

Otherwise, we have $\Delta_k > \tau_0 \epsilon / \beta$ for some $k = m, m+1, ..., l$, and consequently

$$f_m - f_{l+1} \geq \eta c_1 \tau_0^2 \epsilon^2 / \beta. \tag{3.18}$$

Since the sequence $\{f_k\}_{k=0}^{\infty}$ is decreasing and bounded from below, we have

$$f_k \searrow f^* \tag{3.19}$$

for some $f^* > -\infty$. Therefore, using (3.17) and (3.18), we can write

$$\begin{aligned} f_m - f^* &\geq f_m - f_{l+1} \\ &\geq \eta c_1 \tau_0 \epsilon^2 \min\Big(\frac{\tau_0}{\beta}, \frac{1}{\beta_1}\Big) \\ &= \frac{1}{4} \eta c_1 \tau_0 \min\Big(\frac{\tau_0}{\beta}, \frac{1}{\beta_1}\Big) \|g_m\|^2. \end{aligned}$$

By rearranging this expression, we obtain

$$\|g_m\|^2 \leq \Big(\frac{1}{4} \eta c_1 \tau_0 \min\Big(\frac{\tau_0}{\beta}, \frac{1}{\beta_1}\Big)\Big)^{-1} (f_m - f^*),$$

so from (3.19) we conclude that $\|g_m\| \to 0 (m \to \infty))$, giving the result. □

It is worthy to note that Theorem 2.6 is a special case of Theorem 3.6.

4 Some New Trust Region Methods

In this section we propose several computable trust region methods which have global convergence. From the previous section, we can summarize the conclusion as follows.

(a) For $p_k = p_k^l = \tau_k d_k$ with d_k satisfying (3.4) and τ_k satisfying (3.6), we have for $\eta = 0$ that

$$\liminf_{k\to\infty} \|g_k\| = 0, \tag{4.1}$$

and for $\eta \in (0, \frac{1}{4})$ that

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{4.2}$$

(b) If the approximate solution p_k of (2.2) satisfies (2.8) and (3.7), and d_k satisfies (3.4), then, for $\eta = 0$, (4.1) holds; for $\eta \in (0, \frac{1}{4})$, (4.2) holds.

According to the above conclusion, we have two ways to establish some new trust region methods. One way is to find a descent direction d_k satisfying (3.4) and use conclusion (a) to construct some new trust region methods. The other way is to construct some subspace trust region methods.

As we know, the key to using trust region methods is how to solve the subproblem. If d_k satisfies (3.4) then we may obtain a simple trust region method by solving the following simple subproblem

$$\min m_k(\tau d_k) = f_k + \tau g_k^T d_k + \frac{1}{2}\tau^2 d_k^T B_k d_k, \quad s.t. \ \|\tau d_k\| \le \Delta_k,$$

and letting $p_k = \tau_k d_k$ with τ_k satisfying (3.6).

In Algorithm 2.1, the trust region subproblem is replaced by the above subproblem, we can obtain a simple trust region algorithm, denoted by LTR (means Line-search Trust Region method).

Furthermore, given a positive integer m, when $k \ge m$, let

$$Z_k = [d_k, q_1^{(k)}, q_2^{(k)}, ..., q_{m-1}^{(k)}],$$

where $d_k, q_1^{(k)}, q_2^{(k)}, ..., q_{m-1}^{(k)}$ are m vectors in R^n with d_k satisfying (3.4). Set $d = Z_k y$ with $y \in R^m$. Then we can obtain a subproblem

$$\min m_k(Z_k y) = f_k + g_k^T Z_k y + \frac{1}{2} y^T Z_k^T B_k Z_k y, \quad s.t. \ \|Z_k y\| \le \Delta_k, \tag{4.3}$$

where m is substantially smaller than n. This is to say that (4.3) is easier to be solved than (2.2). If y_k is a solution to the above subproblem, then we take $p_k = Z_k y_k$. We can obtain a new trust region method.

We call the corresponding algorithm the subspace trust region method. The matrix Z_k has many special forms, for example, whenever $k \ge m$,

(i) $Z_k = [-g_k, p_{k-1}, ..., p_{k-m+1}]$;

(ii) $Z_k = [-g_k, \gamma_{k-1}, ..., \gamma_{k-m+1}]$, or

(iii) $Z_k = [-g_k, s_{k-1}, ..., s_{k-m+1}]$, where $\gamma_k = g_{k-i+1} - g_{k-i}$ and $s_k = x_{k-i+1} - x_{k-i}$ with $i = 1, 2, ..., m$.

(iv) $Z_k = [d_k, g_{k-1}, g_{k-2}, ..., g_{k-m+1}]$ with d_k satisfying (3.4).

We denote the subspace trust region method by STR (Subspace Trust Region method) in the paper.

Theorem 4.1. *In Algorithm 2.1, $p_k = Z_k y_k$ and y_k is a solution to (4.3). Suppose that $\|B_k\| \le \beta$ for some constant β, f is Lipschitz continuously differentiable and bounded below on the level set $\{x \in R^n | \ f(x) \le f(x_0)\}$. Then, for $\eta = 0$, (4.1) holds; for $\eta \in (0, \frac{1}{4})$, (4.2) holds.*

Proof. It suffices to show that p_k satisfies (2.8) and (3.7). Then, using Theorems 3.4 and 3.6 we can draw the conclusion. It is certain that p_k satisfies (2.8). It needs only to prove that p_k satisfies (3.7).

Construct a vector $\hat{y} = (\hat{y}_1, 0, ..., 0)^T \in R^m$ and solve the following subproblem

$$\min m_k(Z_k\hat{y}) = f_k + g_k^T Z_k\hat{y} + \frac{1}{2}\hat{y}^T Z_k^T B_k Z_k\hat{y}, \quad s.t.\ \|Z_k\hat{y}\| \le \Delta_k.$$

Since $Z_k\hat{y} = \hat{y}_1 d_k$, the subproblem can be changed as

$$\min m_k(\hat{y}_1 d_k) = f_k + \hat{y}_1 g_k^T d_k + \frac{1}{2}\hat{y}_1^2 d_k^T B_k d_k, \quad s.t.\ \|\hat{y}_1 d_k\| \le \Delta_k. \tag{4.4}$$

The problem is completely equivalent to (3.5). By Lemma 3.1 we have

$$m_k(0) - m_k(\hat{y}_1 d_k) \ge -\frac{1}{2}\frac{g_k^T d_k}{\|d_k\|}\min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\|\cdot\|B_k\|}\Big),$$

where $\hat{y}_1$ is a solution to (4.4). Noting that

$$m_k(0) - m_k(\hat{y}_1 d_k) \le m_k(0) - m_k(Z_k y_k) = m_k(0) - m_k(p_k),$$

in which y_k is a solution to (4.3), we have

$$m_k(0) - m_k(p_k) \ge -c_1\frac{g_k^T d_k}{\|d_k\|}\min\Big(\Delta_k, -\frac{g_k^T d_k}{\|d_k\|\cdot\|B_k\|}\Big)$$

with $c_1 = \frac{1}{2}$. This shows that p_k also satisfies (3.7). Thus, Theorems 3.4 and 3.6 hold. The proof is completed. □

Moreover, we can simplify the subproblem (4.3) into the following subproblem

$$\min m_k(Z_k y) = f_k + g_k^T Z_k y + \frac{1}{2}y^T Z_k^T B_k Z_k y, \quad s.t.\ \|y\| \le \frac{\Delta_k}{\|d_k\|}, \tag{4.5}$$

Theorem 4.2. *In Algorithm 2.1, $p_k = Z_k y_k$ and y_k is a solution to (4.5). Suppose that $\|B_k\| \le \beta$ for some constant β, f is Lipschitz continuously differentiable and bounded below on the level set $\{x \in R^n |\ f(x) \le f(x_0)\}$. Then, for $\eta = 0$, (4.1) holds; for $\eta \in (0, \frac{1}{4})$, (4.2) holds.*

Proof. The proof is similar to that of Theorem 4.1 and omitted here. □

The subproblem (4.5) is easier to be solved than (4.3) because m is far smaller than n.

In order to solve large scale optimization problems by using the trust region method, we need to avoid the storage and calculation of some matrices such as B_k. The subproblem of trust region method can be changed into

$$\min m_k(p) = f_k + g_k^T p + \frac{1}{2}L_k\|p\|^2, \quad s.t.\ \|p\| \le \Delta_k, \tag{4.6}$$

where L_k is a parameter that approximates to the Lipschitz constant of the gradient of objective functions $f(x)$. The subproblem is simple and easy to be solved in practical computation. Generally, we require L_k to satisfy

$$0 < L_k \le \beta. \tag{4.7}$$

In fact, we can solve (4.6) and obtain

$$p_k = \begin{cases} -(1/L_k)g_k, & \text{if } \frac{\|g_k\|}{L_k} \le \Delta_k; \\ -(\Delta_k/(L_k\|g_k\|))g_k, & \text{otherwise.} \end{cases} \tag{4.8}$$

The corresponding trust region algorithm is denoted by STR(4.6).

In practical computation, we can obtain some estimations of L_k. Firstly, for $k \ge 2$, we can take

$$L_k = \frac{\|g_k - g_{k-1}\|}{\|x_k - x_{k-1}\|}. \tag{4.9}$$

Secondly, we may take L_k to be a solution to the following minimization problem

$$\min_{L\in R^1} \|Ls_{k-1} - \gamma_{k-1}\|,$$

where $s_{k-1} = x_k - x_{k-1}$ and $\gamma_{k-1} = g_k - g_{k-1}$, so the solution is

$$L_k = \frac{s_{k-1}^T \gamma_{k-1}}{\|s_{k-1}\|^2}; \tag{4.10}$$

or by solving

$$\min_{L\in R^1} \|\frac{1}{L}s_{k-1} - \gamma_{k-1}\|$$

we obtain

$$L_k = \frac{\|\gamma_{k-1}\|^2}{s_{k-1}^T \gamma_{k-1}}. \tag{4.11}$$

Furthermore, let $D_k \in R^{n\times n}$ be a diagonal matrix or a Hessenberg matrix that is an approximation to the Hessian $G(x_k)$ of $f(x)$ at the point x_k, we may consider the subproblem

$$\min m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T D_k p, \quad s.t.\ \|p\| \le \Delta_k. \tag{4.12}$$

Denote $D_k = diag(d_{11}^{(k)}, d_{22}^{(k)}, ..., d_{nn}^{(k)})$ or denote $D_k = (d_{ij}^{(k)})_{n\times n}$, where $d_{ij}^{(k)} = 0$ whenever $|i-j| > l$ (we call it a Hessenberg matrix with band l; if $l = 0$ then the Hessenberg matrix reduces to a diagonal matrix). We can estimate D_k for $k \ge 2$ by solving the following minimization problem

$$\min \|Ds_{k-1} - \gamma_{k-1}\|, s.t.\ |d_{ii}^{(k)}| \le \beta,\ i = 1, 2, ..., n, \tag{4.13}$$

where $\delta_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

The subproblem (4.12) should be easier to be solved than (2.2) in some sense. We denote the algorithm with subproblem (4.12) by STR(4.12).

In the following, we shall choose some test problems to implement the new versions of trust region method, LTR, STR(4.6) and STR(4.12). The problems and their initial iterative points are from the literature ([11]) and denote test problems 1-18 as the same in the literature. The BFGS formula is used to modify the matrix sequence $\{B_k\}$ and $d_k = -B_k^{-1} g_k$ in LTR if B_k^{-1} is available.

We use $\bar{\Delta} = 10^6$, $\delta_0 = 0.5$, $\eta = 0.12$, $L_0 = 0.01$, $\beta = 1000$ and L_k defined by (4.9), (4.10) or (4.11) (denote Algorithms STR(4.9), STR(4.10) and STR(4.11)) in the implementation of STR(4.6) and the stopping criteria is $\|g_k\| \le 10^{-11}$. The number of iterations and total CPU time are listed in Table 1. In Algorithms STR(4.9), STR(4.10) and STR(4.11), if $L_k \in [L_0, \beta]$ then we take $L_k = L_k$ otherwise we take $L_k = L_0$ when $L_k < L_0$ and $L_k = \beta$ when $L_k > \beta$.

Table 1: The number of iterations and total CPU time

P	n	LTR	STR(4.9)	STR(4.10)	STR(4.11)	STR(4.12)
1	2	26	52	28	54	36
2	2	18	42	27	46	35
3	2	12	22	16	25	20
4	2	15	34	17	33	28
5	2	21	37	26	38	32
6	2	15	35	25	39	28
7	3	36	81	42	83	48
8	3	45	84	57	89	73
9	3	37	76	64	82	65
10	3	23	46	38	56	45
11	3	39	69	51	73	62
12	3	37	67	62	65	63
13	4	47	84	63	86	81
14	4	28	123	74	108	112
15	4	69	127	87	131	116
16	4	19	67	46	69	53
17	5	79	135	84	123	121
18	6	31	57	55	59	43
CPU	-	46s	84s	75s	88s	82s

Numerical results in Table 1 show that some modified trust region methods are effective in practical computation. Specifically, the best modification seems to be LTR because it uses less total CPU time than other similar methods mentioned in this paper. However, LTR needs to memorize and compute some matrices in its implementation. STR(4.10) seems to be the best one in STR methods because it takes only 75 seconds for solving 18 problems.

Moreover, the subproblems of the modified trust region methods are easier to be solved than those of the original trust region methods. This makes the modified trust region methods implementable, available, and effective in practical computation.

5 Conclusions

In this paper, we developed some new properties of the trust region method for unconstrained optimization by generalizing Cauchy point to a general form. These new properties enable us to design some new and effective trust region methods. Moreover, we proposed several simple and implementable trust region algorithms in which the subproblem was simple and easy to be solved. Preliminary numerical results showed that some new trust region algorithms were available and efficient in practical computation. In particular, the subspace trust region method should be a promising algorithm because it may be closely related to line search methods.

For future research we should choose different d_k to construct new trust region methods and make the new subproblems easier to be solved than the original subproblems. Furthermore, by using the idea proposed in this paper, we can investigate the relationship between the line search method and the trust region method and design some new hybrid and robust optimization methods.

Acknowledgement

The first author thanks Professor Ya-xiang Yuan for his excellent guidance and interesting teaching on optimization methods in Chinese Academy of Sciences. The authors would like to thank the two anonymous referees and the editor for their helpful comments and suggestions.

References

[1] R.H. Byrd, H.F. Khalfan and R.B. Schnabel, Analysis of a symmetric rank-one trust region method, *SIAM J. Optim.* 6 (1996) 1025–1039.

[2] R.H. Byrd, J. C. Gilbert and J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Math. Program.* 89 (2000) 149–185.

[3] A.R. Conn, N.I. M. Gould and P.L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia PA, 2000.

[4] G. Corradi, A trust region algorithm for unconstrained optimization, *J. Comput. Math.* 65 (1997) 109–119.

[5] D. Calvetti and L. Reichel, Gauss quadrature applied to trust region computations, *Numer. Algorithms* 34 (2003) 85–102.

[6] Y.H. Dai and D.C. Xu, A new family of trust region algorithms for unconstrained optimization, *J. Comput. Math.* 21 (2003) 221–228.

[7] L. Hei, A self-adaptive trust region algorithm, *J. Comput. Math.* 21 (2003) 229–236.

[8] D.J. Higham, Trust region algorithms and timestep selection, *SIAM J. Numer. Anal.* 37 (1999) 194–210.

[9] M. Marazzi and J. Nocedal, Wedge trust region methods for derivative free optimization, *Math. Program.* 91 (2002) 289–305.

[10] J.J. Moré and D.C. Sorensen, On the use of directions of negative curvature in a modified Newton method, *Math. Program.* 16 (1979) 1–20.

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

[12] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer-Verlag, New York, 1999.

[13] Q. Ni, A globally convergent method of solving asymtotes with trust region technique, *Optim. Methods Soft.* 18 (2003) 283–297.

[14] M.J.D. Powell, Convergence properties of a class minimization algorithms, in *Nonlinear Programming 2*, O.L. Mangasarian, R.R. Meyer and S.M. Robinson (eds.), Academic Press, New York, 1975, pp. l–27.

[15] M.J.D. Powell, On the global convergence of trust region algorithms for unconstrained optimization, *Math. Program.* 29 (1984) 297–303.

[16] L. Qi and J. Sun, A trust region algorithm for minimization of locally Lipschitzian functions, *Math. Program.* 66 (1994) 25–45.

[17] M. Rojas, S.A. Santos and D.C. Sorensen, A new matrix-free algorithm for the large scale trust region subproblem, *SIAM J. Optim.* 11 (2000) 611–646.

[18] J. Sun, On piecewise quadratic Newton and trust region problems, *Math. Program.* (Series B) 76 (1997) 451–467.

[19] W.Y. Sun, Nonmonotone trust region method for solving optimization problems, *Appl. Math. Comput.* 156 (2004) 159–174.

[20] G.A. Schultz, R.B. Schnabel and R.H. Byrd, A family of trust-region-based algorithms for unconstrained minimization with strong global convergence, *SIAM J. Namer. Anal.* 22 (1985) 47–67.

[21] Z.J. Shi, Convergence of line search methods for unconstrained optimization, *Appl. Math. Comput.* 157 (2004) 393–405.

[22] Z.J. Shi and J. Shen, Convergence of descent method without line search, *Appl. Math. Comput.* 167 (2005) 94–107.

[23] C.X. Xu and J.Z. Zhang, Scaled optimal path trust-region algorithm, *J. Optim. Theory Appl.* 102 (1999) 127–146.

[24] Y. Yuan, On the convergence of a new trust region algorithm, *Numer. Math.* 70 (1995) 515–539.

[25] Y. Yuan, *A review of trust region algorithms for optimization*, ICIAM 99 (Edinburgh) 271-283, Oxford Univ. Press, Oxford 2000.

[26] Y. Yuan, On the convergence of trust region algorithms, Mathematics *Numerica Sinica* 16 (1996) 333–346.

[27] Y. Yuan and W.Y. Sun, *Optimization Theory and Methods*, Science Press of China, 1997 (in Chinese).

[28] J.Z. Zhang and C.X. Xu, Trust region dogleg path algorithms for unconstrained minimization, *Annals of Operations Research* 87 (1999) 407–418.

Manuscript received 23 May 2006
revised 5 August 2006, 2 September 2006
accepted for publication 7 September 2007

Zhen-Jun Shi
College of Operations Research and Management
Qufu Normal University, Rizhao, Shandong 276826, P.R.China, and
Department of Computer & Information Science
University of Michigan, Dearborn, MI 48128, USA
E-mail address: zjshi@qrnu.edu.cn, zjshi@umd.umich.edu

Jie Shen
Department of Computer & Information Science
University of Michigan, Dearborn, MI 48128, USA
E-mail address: shen@umd.umich.edu

Jinhua Guo
Department of Computer & Information Science
University of Michigan, Dearborn, MI 48128, USA
E-mail address: jinhua@umd.umich.edu