COMPUTATIONAL & APPLIED MATHEMATICS
Volume 29, N. 2, pp. 89–106, 2010

ISSN 0101-8205
www.scielo.br/cam

Using truncated conjugate gradient method in trust-region method with two subproblems and backtracking line search*

MINGYUN TANG[1] and YA-XIANG YUAN[2]

[1]College of Science, China Agricultural University, Beijing, 100083, China
[2]State Key Laboratory of Scientific Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing, 100190, China

E-mails: tangmy@lsec.cc.ac.cn / yyx@lsec.cc.ac.cn

Abstract. A trust-region method with two subproblems and backtracking line search for solving unconstrained optimization is proposed. At every iteration, we use the truncated conjugate gradient method or its variation to solve one of the two subproblems approximately. Backtracking line search is carried out when the trust-region trail step fails. We show that this method have the same convergence properties as the traditional trust-region method based on the truncated conjugate gradient method. Numerical results show that this method is as reliable as the traditional one and more efficient in respect of iterations, CPU time and evaluations.

Mathematical subject classification: Primary: 65K05; Secondary: 90C30.

Key words: truncated conjugate gradient, trust-region, two subproblems, backtracking, convergence.

---

1 Introduction

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

#CAM-99/09. Received: 28/IV/09. Accepted: 30/VI/09.
*Partially supported by Chinese NSF grant 10831006 and CAS grant kjcx-yw-s7.

where f is a real-valued twice continuously differentiable function, which we assume bounded below. Unconstrained optimization problems are essential in mathematical programming because they occur frequently in many real-world applications and the methods for such problems are fundamental in the sense that these methods can be either directly applied or extended to optimization problems with constraints. There are many effective algorithms designed for unconstrained optimization problems (see [6, 13]), most of the algorithms can be classified into two categories: line search algorithm and trust region algorithms.

Trust-region method is efficient for solving (1). It has mature framework, strong convergence properties and satisfactory numerical results (see [4]). However, sometimes the trust-region step may be too conservative, especially when the objective function has large "convex basins". The standard trust-region technique may require quit a few iterations to stretch the trust region in order to contain the local minimizer. Thus, it is natural for us to consider modifying the standard trust region method to give an new algorithm which can maintain the convergence properties of the standard trust-region method and need less computational cost.

In the previous research [12], we obtained a trust-region method with two subproblems and backtracking line search. A subproblem without trust-region constraint was introduced into the trust-region framework in order to use the unit stepsize Newton step. This unconstrained subproblem normally gives a longer trail step, consequently it is likely that the overall algorithm will reduce the computational cost. Based on the information in the previous iterations, either the trust-region step or the Newton step is used. Moreover, the idea of combining trust-region and line search techniques (see [8]) is also used in that algorithm. The algorithm inherits the convergence result of traditional trust-region method and gives better performance by making good use of the Newton step and backtracking line search. However, as the exact minimizer of the trust-region subproblem is need and the Cholesky factorization of the Hessian matrix is used to decide whether the model function is convex, that algorithm is obviously not fit for large problems.

Therefore, in this paper we propose a new trust-region algorithm with two subproblems using truncated conjugate gradient method and its variation to solve the subproblems. Our method can be regarded as a modification to the standard trust region method for unconstrained optimization in such a way that the Newton step can be taken as often as possible. A slightly modified truncated conjugate gradient method is used to compute the Newton step. The global convergence and local superlinear convergence results of the algorithm are also given in the paper.

The paper is organized as follows. In the next section, we give the framework of the method and describe our new algorithm. The convergence properties are presented in Section 3 and the numerical results are provided in Section 4. Some conclusions are given in Section 5.

2 The algorithm

First we briefly review the framework of trust-region method with two subproblems and backtracking line search.

One of the two subproblems is the trust-region subproblem. At the current iteration x_k, the trust-region subproblem is

$$\begin{cases} \min\limits_{s\in\mathbb{R}^n} & Q_k(s) = g_k^T s + \frac{1}{2} s^T H_k s, \\ s.t. & \|s\| \le \Delta_k \end{cases}, \tag{1}$$

where $Q_k(s)$ is the approximate model function of $f(x)$ within the trust-region, $g_k = g(x_k) = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$ or an approximation of $\nabla^2 f(x_k)$. We usually choose Q_k to be the first three terms of the Taylor expansion of the objective function $f(x)$ at x_k with the constant term $f(x_k)$ being omitted as this term does not influence the iteration process.

Another subproblem is defined by

$$\min_{s\in\mathbb{R}^n} Q_k(s) = g_k^T s + \frac{1}{2} s^T H_k s, \tag{2}$$

where g_k, H_k has the same meaning as in (2). Since in this subproblem we do not require the trust region constraint, we call it unconstrained subproblem.

In the ideal situation, the unconstrained subproblem should be used when the model function is convex and gives an accurate approximation to the objective function. Define

$$\rho_k = \frac{f(x_k) - f(x_k + s)}{Q_k(0) - Q_k(s)}. \tag{3}$$

The ratio ρ_k is used by trust region algorithms to decide whether the trial step is acceptable and how to update the trust-region radius. In the method given in [12], we also use the value of ρ_k and the positive definiteness of $\nabla^2 f(x_k)$ to decide the model choice since we solve the trust-region subproblem exactly. In this paper, we use the truncated conjugate gradient method (see [1, 10]) to compute a minimizer of the trust-region subproblem approximately, as Cholesky factorization cannot be used for large scale problems. Now, we consider how to compute the unconstrained model minimizer approximately.

Consider using the conjugate gradient method to solve the subproblem (3). The subscript i denotes the interior iteration number. If we do not know whether our quadratic model is strictly convex, precautions must be taken to deal with non-convexity if it arises. Similarly to the analysis of the truncated conjugate gradient method (see [4]), if we minimize Q_k without considering whether or not $\nabla^2 f(x_k)$ is positive definite, the following two possibilities might arise:

(i) the curvature $\langle p_i, Hp_i \rangle$ is positive at each iteration. This means the current model function is convex along direction p_i, as we expect. In this case, we just need to continue the iteration of the conjugate gradient method.

(ii) $\langle p_i, Hp_i \rangle \leq 0$. This means the model function is not strictly convex. Q_k is unbounded from below along the line $s_i + \sigma p_i$. The unconstrained subproblem is obviously not fit for reflecting the condition of the objective function around the current iteration now. So we should add the trust-region constraint and minimize Q_k along $s_i + \sigma p_i$ as much as we can while staying within the trust region. In this case, what we need to do is finding the positive root of $\|s_i + \sigma p_i\| = \Delta$.

In order to avoid conjugate gradient iterations that make very little progress in the reduction of the model quadratical function, the iteration are also terminated if one of the following conditions

$$\|\nabla Q_k(s_j)\| \leq 10^{-2}\|\nabla Q_k(0)\|, \tag{4}$$

$$\big[Q_k(s_{j-1}) - Q_k(s_j)\big] \leq 10^{-2}\big[Q_k(0) - Q_k(s_j)\big] \tag{5}$$

is satisfied (see [9]). The iterations are also terminated if the theoretical upper bound n is reached.

Now, we can give an algorithm for solving the unconstrained subproblem approximately. It is a variation of the truncated conjugate gradient method.

Algorithm 2.1

Step 1 *Initialization* *Set* $s_0 = 0$, $v_0 = g_0 = \nabla_x f(x_k)$, $p_0 = -g_0$, $curq = 0$, $preq = 1$, $kappag = 0.01$, $\varepsilon_k = \min(kappag, \sqrt{\|g_0\|})$, $itermax = n$.

Step 2 *While* $i \leq itermax$

if $preq - curq \leq kappag * (-curq)$, *stop.*

$\kappa_i = p_i^T H p_i$,

if $\kappa_i \leq 0$, *then*

info:= 1, *if* $\|s_i\| \geq \Delta_k$ *then stop, else compute* σ *as the positive root of* $\|s_i + \sigma p_i\| = \Delta_k$, $s_{i+1} = s_i + \sigma p_i$, *stop.*

end if

$\alpha_i = \langle g_i, v_i \rangle / \kappa_i$,

$s_{i+1} = s_i + \alpha_i p_i$,

update $preq$, $curq$, g_{i+1},

if $\|g_{i+1}\| / \|g_0\| \leq \varepsilon_k$, *stop.*

$v_{i+1} = g_{i+1}$,

$\beta_i = \langle g_{i+1}, v_{i+1} \rangle / \langle g_i, v_i \rangle$,

$p_{i+1} = -v_{i+1} + \beta_i p_i$,

$i = i + 1$,

goto step 2.

The above modification of the conjugate gradient method can deal with negative curvature directions. Such technique is also discussed in [1] as well. In the above algorithm, if *info* equals 1, the current model function is not convex. It is easy to see that the computation cost in each iteration of the above algorithm is mainly one matrix-vector multiplication. Thus, it is very likely that the above algorithm is faster than solving $H_k s = -g_k$ by carrying out the Cholesky factorization of H_k directly.

We now describe the algorithm of using truncated conjugate gradient method in the trust-region method with two subproblems and backtracking line search. We use ρ_k and flag *info* to decide the model choice. If the value of ρ_k of an unconstrained subproblem is smaller than a positive constant η_2 or $info = 1$, we may consider that the unconstrained model is not proper and choose the trust-region subproblem in the next iteration. We take the unconstrained model if the ratio ρ_k of the trust-region trail step is bigger than a constant β ($\beta \to 1$ and $\beta < 1$) in 2 consecutive steps. The overall algorithm is given as follows.

Algorithm 2.2 (A trust region method with two subproblems)

Step 1 Initialization.

An initial point x_0 and an initial trust-region radius $\Delta_0 > 0$ are given. The stopping tolerance ε is given. The constants η_1, η_2, γ_1, γ_2 and β are also given and satisfy $0 < \eta_1 \leq \eta_2 < \beta < 1$ and $0 < \gamma_1 < 1 \leq \gamma_2$. Set $k := 0$, btime $:= 0$ and $f_{min} := f(x_0)$. Set flag $TR_0 := 0$, info $:= 0$.

Step 2 Determine a trial step.

if $TR_k = 0$, then compute s_k by Algorithm 2.1;

else use truncated CG method to obtain s_k.

$x_t := x_k + s_k$.

Step 3 Backtracking line search.

$f_t := f(x_t)$.

if $f_t < f_{min}$ go to step 4;

else if $TR_k = 1$, then carry out backtracking line search;

else then

$TR_{k+1} := 1$, btime $:= 0$, $k := k + 1$, go to step 2.

Step 4 Acceptance of the trial point and update the flag TR_{k+1} and the trust-region radius.

$x_{k+1} := x_t$. $f_{min} := f(x_t)$.

Compute ρ_k *according to* (4).

Update Δ_{k+1} :

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k & if \;\; TR_k = 1 \;\; and \;\; \rho_k < \eta_1 \,, \\ & or \;\; TR_k = 0 \; and \; \rho_k < \eta_1 \;\; and \;\; \|s_k\| \le \Delta_k \,; \\ \gamma_2 \Delta_k & if \;\; TR_k = 1 \;\; and \;\; \rho_k \ge \eta_2 \,, \\ & or \;\; TR_k = 0 \; and \; \rho_k \ge \eta_2 \;\; and \;\; info = 1 \,; \\ \Delta_k & otherwise. \end{cases}$$

Update btime and TR_{k+1} :

$$btime = \begin{cases} 0 & if \;\; TR_k = 1 \;\; and \;\; \rho_k \le \beta \,, \\ & or \;\; TR_k = 0 \;\; and \;\; \rho_k \ge \eta_2 \;\; and \\ & \;\; info = 1 \,, \\ & or \;\; TR_k = 0 \;\; and \;\; 0 < \rho_k < \eta_2 \,; \\ btime + 1 & if \;\; TR_k = 1 \;\; and \;\; \rho_k > \beta. \end{cases}$$

$$TR_{k+1} = \begin{cases} 0 & if \;\; btime = 2 \,; \\ 1 & if \;\; TR_k = 0 \;\; and \;\; \rho_k \ge \eta_2 \;\; and \;\; info = 1 \,, \\ & or \;\; TR_k = 0 \;\; and \;\; 0 < \rho_k < \eta_2 \,; \\ TR_k & otherwise. \end{cases}$$

if btime $= 2$, *btime* $:= 0$.

$k := k + 1$.

go to step 2.

In the above algorithm, backtracking line search is carried out using the same formula as in [12]. We try to find the minimum positive integer i such that $f(x_k + \alpha^i s) < f(x_k)$, where $\alpha \in (0, 1)$ is a positive constant (see [8]). The step size α is computed by polynomial interpolation since $f(x_k)$, $g(x_k)$, $\nabla^2 f(x_k)$ and $f(x_k + s)$ are all known. Denote $q = \frac{1}{2} s^T \nabla^2 f(x_k) s$, then

$$\alpha = -\frac{g_k^T s}{q + \sqrt{(q^2 - 3 g_k^T s (f(x_k + s) - q - g_k^T s - f(x_k))}} \tag{6}$$

(choose $\alpha = -g_k^T s / s^T \nabla^2 f(x_k) s$ or the result of truncated quadratic interpolation when the denominator equals to zero). Set $\alpha_k = \max[0.1, \alpha]$, in case that

α_k is too small. It is obvious that to evaluate the objective function on two close points is a waste of computational cost and available information. The idea of discarding small steps computed as minimizers of interpolation functions was also explored in [2] and [3].

3 Convergence

In this section we present the convergence properties of the algorithm given in the previous section.

In our algorithm, if the unconstrained subproblem is chosen and the Hessian matrix is not positive definite, the trial step will be truncated on the boundary of trust-region just the same as truncated conjugate gradient method. So the difference arises when the model function is convex and the trail step is large, which provides more decrease of the model function. Thus in fact the unconstrained model is used only when the model function is convex. The proof of our following theorem is similar to that of Theorem 3.2 of Steihaug [11] and Powell [10].

Theorem 3.1. *Suppose that f in* (1) *is twice continuously differentiable and bounded below and the norm of Hessian matrix is bounded. Let ε_k be the relative error in the truncated conjugate gradient method and the Algorithm 2.1. Iteration $\{x_k\}$ is generated by the Algorithm 2.2. If $\varepsilon_k \leq \varepsilon < 1$, then*

$$\liminf_{k\to\infty} \|g(x_k)\| = 0. \tag{1}$$

Proof. Since we have

$$Q_k(s_k) \leq \min\left\{Q_k(-\sigma g(x_k)) : \|\sigma g(x_k)\| \leq \|s_k\|\right\} \tag{2}$$

no matter the trail step s_k is computed by the trust-region subproblem or the unconstrained subproblem, it follows from Powell's result [10] that

$$-Q_k(s_k) \geq c_1 \|g(x_k)\| \min\left\{\|s_k\|, \frac{\|g(x_k)\|}{\|H_k\|}\right\}, \tag{3}$$

with $c_1 = \dfrac{1}{2}$.

We prove the theorem by contradiction. If the theorem were not true, we can assume that

$$\|g(x_k)\| \geq \delta \text{ for all } k. \tag{4}$$

Thus, due to (4) and the boundedness of $\|H_k\|$, there exists a positive constant $\bar{\delta}$ such that

$$-Q_k(s_k) \geq \bar{\delta} \min\left\{\|s_k\|, 1\right\}. \tag{5}$$

First we show that

$$\sum_{k\geq 0} \|s_k\| < +\infty\,. \tag{6}$$

Define the following two sets of indices:

$$S = \left\{k : \rho_k \geq \eta_1\right\}, \tag{7}$$

$$U = \left\{k : TR_k = 0\right\}. \tag{8}$$

Since f is bounded below, we have

$$\begin{aligned}
+\infty \quad > \quad & f_0 - \min f(x) \geq \sum_{k=1}^{\infty} \left[f(x_k) - f(x_{k+1})\right] \qquad (9)\\
\geq \quad & \sum_{k\in S} \left[f(x_k) - f(x_{k+1})\right] \geq \eta_1 \sum_{k\in S} -Q(s_k)\\
\geq \quad & \eta_1 \bar{\delta} \sum_{k\in S} \min\left\{\|s_k\|, 1\right\},
\end{aligned}$$

which shows that

$$\sum_{k\in S} \|s_k\| < +\infty. \tag{10}$$

Hence there exists k_0 such that

$$\|s_k\| \geq \Delta_k \qquad \forall k \geq k_0,\ k \in S \tag{11}$$

because $\|H_k s_k + g_k\| \geq \frac{1}{2}\delta$ for all sufficiently large k. This shows that

$$\Delta_{k+1} \leq \gamma_2 \Delta_k \leq \gamma_2 \|s_k\| \qquad \forall k \geq k_0,\ k \in S. \tag{12}$$

First we consider the case that U is a finite set. In this case, there exists an integer k_1 such that $TR_k = 1$ for all $k \geq k_1$ and Algorithm 2.2 is essentially the standard trust-region algorithm for all large k. Thus

$$\Delta_{k+1} \leq \gamma_1 \Delta_k \qquad \forall k \geq k_1, \ k \notin S. \tag{13}$$

Let $k_2 = \max\{k_0, k_1\}$, we have that

$$\begin{aligned}
\sum_{k \geq k_2} \|s_k\| &\leq \sum_{k \geq k_2,\, k \in S} \|s_k\| + \sum_{k \geq k_2,\, k \notin S} \Delta_k \\
&\leq \sum_{k \geq k_2,\, k \in S} \|s_k\| + \frac{\Delta_{k_2} + \gamma_2 \sum_{k \geq k_2,\, k \in S} \Delta_k}{1 - \gamma_1} \\
&\leq \frac{\Delta_{k_2}}{1 - \gamma_1} + \left(1 + \frac{\gamma_2}{1 - \gamma_1}\right) \sum_{k \geq k_2,\, k \in S} \|s_k\| \\
&< +\infty,
\end{aligned} \tag{14}$$

which shows that (6) is true.

Now we consider the case that U has infinitely many elements.

If $k \notin S$, $TR_k = 0$ and k is sufficiently large, we have that $TR_{k+1} = 1$ and

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k & \text{if} \quad \|s_k\| \leq \Delta_k \,, \\ \Delta_k & \text{if} \quad \|s_k\| > \Delta_k \,. \end{cases} \tag{15}$$

while $k \notin S$, $TR_k = 1$, always have $\Delta_{k+1} = \gamma_1 \Delta_k$. Therefore there exists k_3 such that

$$\begin{aligned}
\sum_{k \geq k_3,\, k \notin S} \Delta_k &\leq \left(\Delta_{k_3} + \sum_{k \geq k_3,\, k \in S} \Delta_k\right) \frac{2\gamma_2}{1 - \gamma_1} \\
&\leq \left(\Delta_{k_3} + \sum_{k \geq k_3,\, k \in S} \|s_k\|\right) \frac{2\gamma_2}{1 - \gamma_1}.
\end{aligned} \tag{16}$$

Hence

$$\sum_{k \geq k_3,\, k \notin S} \Delta_k < +\infty \,. \tag{17}$$

Relation (10), (11) and (17) indicate that

$$\sum_{k=1}^{\infty} \Delta_k < +\infty\,, \tag{18}$$

which implies that $\lim_{k\to\infty} \Delta_k = 0$. Therefore, when $k \to +\infty$ and $\|s_k\| \le \Delta_k$, we have

$$\rho_k \to 1. \tag{19}$$

Thus, $\Delta_{k+1} \ge \Delta_k$ if k is sufficiently large and if $\|s_k\| \le \Delta_k$. If $\|s_k\| > \Delta_k$, we know that $TR_k = 0$, our algorithm gives either $\Delta_{k+1} = \Delta_k$ or $\Delta_{k+1} = \gamma_2\Delta_k$. This shows that $\Delta_{k+1} \ge \Delta_k$ for all large k. This contradicts to (18). So (4) must therefore be false, which yields (1). □

The above theorem shows that our algorithm is globally convergent. Furthermore, we can show that our algorithm converges superlinearly if certain conditions are satisfied.

Theorem 3.2. *Suppose that f in (1) in Section 1 is twice continuously differentiable and bounded below and the norm of Hessian matrix is bounded, the iteration $\{x_k\}$ generated by Algorithm 2.2 satisfies $x_k \to x^*$ as $k \to \infty$ and the Hessian matrix $H(x^*)$ of f is positive definite. Let ε_k be the relative error in the truncated conjugate gradient method and Algorithm 2.1. If $\varepsilon_k \to 0$ then $\{x_k\}$ converges superlinearly, i.e.,*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \tag{20}$$

Proof. Because $x_k \to x^*$ and $H(x^*) > 0$, there exists k_1 such that $\|H^{-1}(x_k)\| \le 2\|H^{-1}(x^*)\|$ for all $k \ge k_1$. Therefore the Newton step $s_k^N = -H^{-1}(x_k)g(x_k)$ satisfies that

$$\|s_k^N\| \le 2\|H^{-1}(x^*)\|\,\|g(x_k)\| \tag{21}$$

for all $k \ge k_1$. Therefore, no matter s_k generated by our algorithm is a trust-region step or a truncated Newton step, we have that

$$\|s_k\| \le 2\|H^{-1}(x^*)\|\,\|g(x_k)\|, \qquad \forall k \ge k_1. \tag{22}$$

Our previous theorem implies that $\|g(x_k)\| \to 0$. Inequality (22) shows that

$$\lim_{k\to\infty} \rho_k = 1. \tag{23}$$

Consequently, $x_{k+1} = x_k + s_k$ and $\Delta_{k+1} \geq \Delta_k$ for all sufficiently large k. Consequently, $\|s_k\| < \Delta_k$ for all sufficiently large k. Namely s_k is an inexact Newton step for all large k, which indicates that

$$\|g(x_k) + H_k s_k\| / \|g(x_k)\| \leq \varepsilon_k, \tag{24}$$

for all sufficiently large k. Relation (24) shows that

$$\begin{aligned} \lim_{k\to\infty} \frac{\|g(x_{k+1})\|}{\|g(x_k)\|} &= \lim_{k\to\infty} \frac{\|g(x_k + s_k)\|}{\|g(x_k)\|} \\ &= \lim_{k\to\infty} \frac{\|g(x_k) + H_k s_k\| + o(\|s_k\|)}{\|g(x_k)\|} \\ &= \lim_{k\to\infty} \frac{\|g(x_k) + H_k s_k\|}{\|g(x_k)\|} \\ &\leq \lim_{k\to\infty} \varepsilon_k = 0. \end{aligned} \tag{25}$$

Now, (20) follows from the fact that $H(x^*) > 0$ and $x_k \to x^*$. □

4 Numerical results

In this section we report numerical results of our algorithm given in Section 2, and we also compare our algorithm with traditional trust region algorithm as given in [6, 13]. Test problems are the 153 unconstrained problems from the CUTEr collection (see [7]). The names and dimensions of the problems are given in Tables 1-3.

The starting point and the exact first and second derivatives supplied with the problem were used. Numerical tests were performed in double precision on a Dell OptiPlex 755 computer (2.66 GHz, 1.96 GB of RAM) under Linux (fedora core 8) and the gcc compiler (version 4.2.3) with default options. All attempts to solve the problems are limited to a maximum of 1000 iterations or 1 hour of CPU time. The choice of the parameters do not have a uniform standard and the

Problem	n	Problem	n	Problem	n
AKIVA	2	CURLY10	1000	DJTL	2
ALLINITU	4	CURLY20	1000	DQDRTIC	1000
ARGLINA	100	CURLY30	1000	DQRTIC	1000
ARGLINB	100	DECONVU	61	EDENSCH	2000
ARGLINC	10	DENSCHNA	2	EG2	1000
ARWHEAD	1000	DENSCHNB	2	EIGENALS	110
BARD	3	DENSCHNC	2	EIGENBLS	110
BDQRTIC	1000	DENSCHND	2	EIGENCLS	30
BEALE	2	DENSCHNE	2	ENGVAL1	1000
BIGGS6	6	DENSCHNF	2	ENGVAL2	3
BOX3	3	DIXMAANA	1500	ERRINROS	50
BRKMCC	2	DIXMAANB	1500	EXPFIT	2
BROWNAL	10	DIXMAANC	300	EXTROSNB	10

Table 1 – Test problems and corresponding dimensions.

parameters are not sensitive to the algorithm. So we choose the common values as (for example, see [6, 13]) $\gamma_1 = 0.25$, $\gamma_2 = 2$, $\eta_1 = 0.1$, $\eta_2 = 0.75$, $\beta = 0.9$, $\Delta_0 = 1$. The truncated conjugate gradient method (see [11]) is used to solve the trust-region subproblem. Both algorithms stop if $\|\nabla f(x_k)\| \leq 10^{-6}$.

Our algorithm solved 125 problems out of the 153 CUTEr test problems, while the traditional trust-region method solved 120 problems. Failure often occurs because the maximal iteration number is reached. Thus, we found that the new algorithm is as reliable as the traditional one.

Both algorithms fail on the same set of 27 problems. For the other 126 problems, the new algorithm needs less iterations on 88 problems. The two algorithms have the same number of iterations on 22 problems and the traditional trust-region method wins on 16 problems. Figure 1 gives the performance profiles (see [5]) for the two algorithms for iterations. Figure 2 gives the performance profiles for CPU times. Considering account inaccuracies in timing, we only compare the CPU times of the 49 test problems whose run-times are longer than 0.1 second and dimensions are larger than 100. The new method takes less time to solve 33 among these 49 problems. Figure 3, 4 and 5 give the performance profiles for function, gradient and Hessian evaluations. Advantage of the new

Problem	n	Problem	n	Problem	n
BROWNBS	2	DIXMAAND	300	FLETCBV2	1000
BROWNDEN	4	DIXMAANE	300	FLETCBV3	10
BROYDN7D	1000	DIXMAANF	300	FLETCHBV	10
BRYBND	1000	DIXMAANG	300	FLETCHCR	100
CHAINWOO	1000	DIXMAANH	300	FMINSRF2	121
CHNROSNB	50	DIXMAANI	300	FMINSURF	121
CLIFF	2	DIXMAANJ	300	FREUROTH	500
COSINE	10	DIXMAANK	15	GENHUMPS	500
CRAGGLVY	100	DIXMAANL	300	GENROSE	100
CUBE	2	DIXON3DQ	1000	GROWTHLS	3
GULF	3	MAQRTBLS	100	SCOSINE	10
HAIRY	2	NONCVXU2	100	SCURLY10	100
HATFLDD	3	NONCVXUN	100	SCURLY20	100
HATFLDE	3	NONDIA	1000	SCURLY30	100
HEART6LS	6	NONDQUAR	1000	SENSORS	10
HEART8LS	8	NONMSQRT	49	SINEVAL	2
HELIX	3	OSBORNEA	5	SINQUAD	500
HIELOW	3	OSBORNEB	11	SISSER	2
HILBERTA	2	OSCIPATH	15	SNAIL	2
HILBERTB	10	PALMER1C	8	SPARSINE	1000
HIMMELBB	2	PALMER1D	7	SPARSQUR	1000
HIMMELBF	4	PALMER2C	8	SPMSRTLS	499
HIMMELBG	2	PALMER3C	8	SROSENBR	1000
HIMMELBH	2	PALMER4C	8	STRATEC	10
HUMPS	2	PALMER5C	6	TESTQUAD	1000
HYDC20LS	99	PALMER6C	8	TOINTGOR	50
INDEF	1000	PALMER7C	8	TOINTGSS	1000
JENSMP	2	PALMER8C	8	TOINTPSP	50
KOWOSB	4	PENALTY1	100	TIONTQOR	50
LIARWHD	1000	PENALTY2	100	TQUARTIC	1000

Table 2 – Test problems and corresponding dimensions.

Problem	n	Problem	n	Problem	n
LOGHAIRY	2	PENALTY3	50	TRIDIA	1000
MANCINO	100	POWELLSG	1000	VARDIM	100
MARATOSB	2	POWER	100	VAREIGVL	50
MEXHAT	2	QUARTC	1000	VIBRBEAM	8
MEYER3	3	ROSENBR	2	WATSON	12
MODBEALE	2000	S308	2	WOODS	1000
MOREBV	1000	SBRYBND	100	YFITU	3
MSQRTALS	100	SCHMVETT	1000	ZANGWIL2	2

Table 3 – Test problems and corresponding dimensions.

algorithm is also shown by total number of evaluations since it is dominative on 77 problems.

It is easy to see from these figures that the new algorithm is more efficient than the traditional trust-region algorithm.

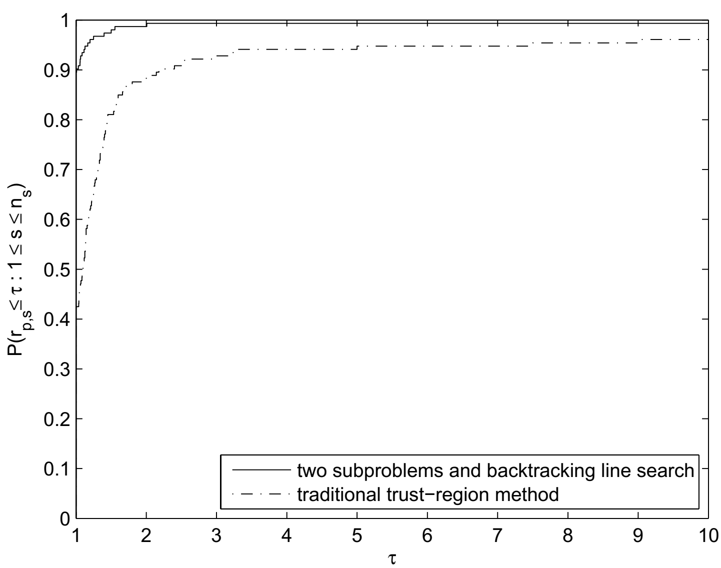

Figure 1 – Performance profiles for iterations.

5 Conclusions

We have proposed a new trust-region algorithm with two subproblems and backtracking line search using truncated conjugate gradient method and its variation

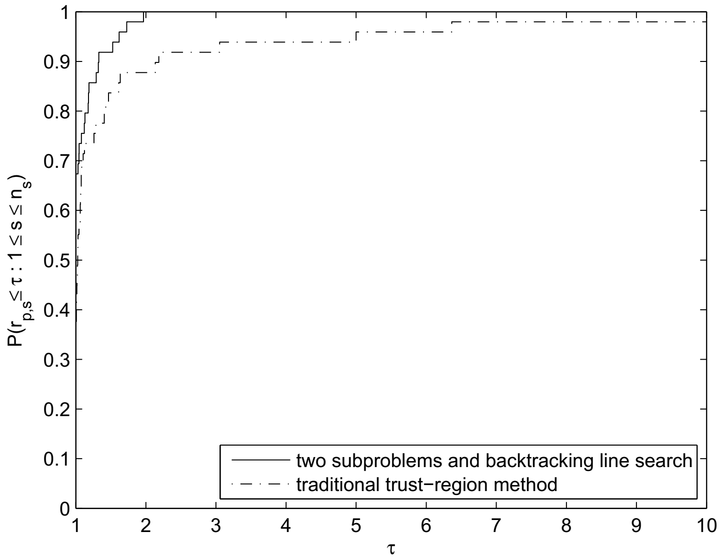

Figure 2 – Performance profiles for CPU times.

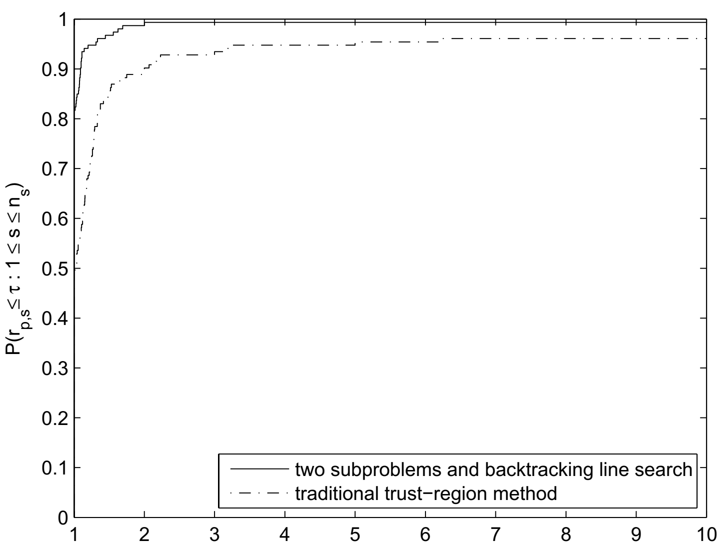

Figure 3 – Performance profiles for function evaluations.

to solve the subproblems. This new algorithm for unconstrained optimization is global convergence and has local superlinear convergence rate when the Hessian matrix of the objective function at the local minimizer is positive definite.

Numerical results on problems from CUTEr collection are also given. The results show that the new algorithm is more efficient than the standard trust-region method in term of the number of iterations and evaluations as well as CPU time.

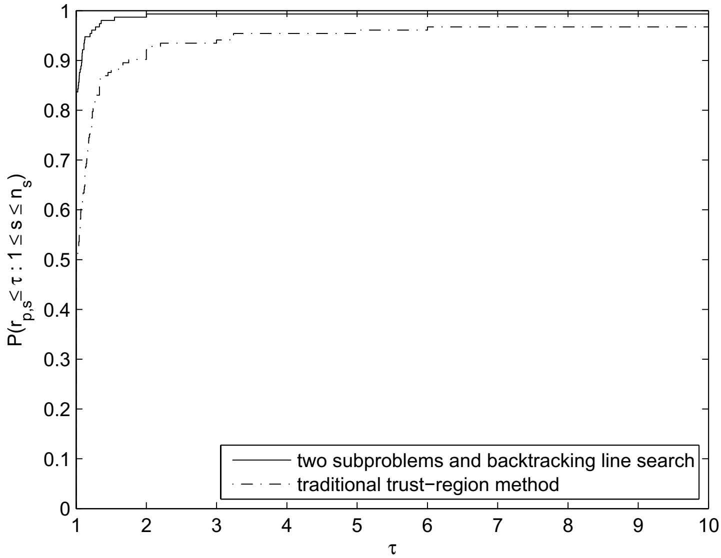

Figure 4 – Performance profiles for gradient evaluations.

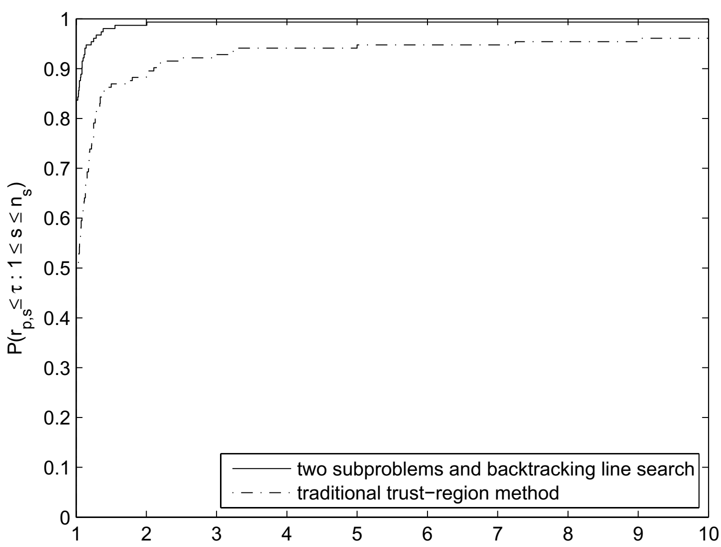

Figure 5 – Performance profiles for Hessian matrix evaluations.

Acknowledgements. The authors would like to thank an anonymous referee for his valuable comments on an earlier version of the paper.

REFERENCES

[1] E.G. Birgin and J.M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients.* Computational Optimization and Applications, **23** (2002), 101–125.

[2] E.G. Birgin, J.M. Martínez and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets.* SIAM Journal on Optimization, **10** (2000), 1196–1211.

[3] E.G. Birgin, J.M. Martínez and M. Raydan, *Algorithm 813: SPG – software for convex-constrained optimization.* ACM Transactions on Mathematical Software, **27** (2001), 340–349.

[4] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Trust-Region Methods.* SIAM, Philadelphia, USA, (2000).

[5] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles.* Mathematical Programming, **91**(2) (2002), 201–213.

[6] R. Fletcher, *Practical Methods of Optimization, Vol. 1, Unconstrained Optimization.* John Wiley and Sons, Chichester, (1980).

[7] N.I.M. Gould, D. Orban and Ph.L. Toint, *CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited.* Transactions of the ACM on Mathematical Software, **29**(4) (2003), 373–394.

[8] J. Nocedal and Y. Yuan, *Combining trust region and line search techniques.* Advances in Nonlinear Programming, (1998), 153–175.

[9] M.J.D. Powell, *The NEWUOA software for unconstrained optimization without derivatives.* Large-Scale Nonlinear Optimization, **83** (2006), 256–297, Springer, USA.

[10] M.J.D. Powell, *Convergence properties of a class of minimization algorithms.* Nonlinear Programming 2, Eds. O.L. Mangasarian, R.R. Meyer and S.M. Robinson, (1975), Academic Press, New York.

[11] T. Steihaug, *The Conjugate Gradient Method and Trust Regions in Large Scale Optimization.* SIAM J. Numer. Anal., **20**(3) (1983), 626–637.

[12] M.Y. Tang, *A trust-region-Newton method for unconstrained optimization.* Proceedings of the 9th National Conference of ORSC, Eds. Y. Yuan, X.D. Hu and D.G. Liu, (2008), Global-Link, Hong Kong.

[13] Y. Yuan, *Computational Methods for Nonlinear Optimization.* Science Press, Beijing, China, (2008). (in Chinese).