Hokkaido Mathematical Journal Vol. 32 (2003) p. 355–369

Combining trust-region and line-search algorithms for minimization subject to bounds[1)]

Xiaojiao TONG and Shuzi ZHOU

(Received January 7, 2002; Revised June 10, 2002)

Abstract. In this paper, we combine the trust-region technique with line searches to develop an iterative method for solving minimization problems subject to bounds. The new method is an extension of the algorithm proposed by Coleman and Li [3]. At each iteration, the solution of the subproblem provides a descent direction of the objective function. If the trial step cannot be accepted by trust-region method, we can use backtracking to find the next iterative point. Compared to the traditional trust-region methods, the new algorithm need not solve the subproblem repeatedly and so it is more economical. Under general conditions, the global convergence of the new algorithm can be proved. A numerical example shows that the new algorithm is promising.

Key words: bound constraints, trust-region method, line search technique, global convergence.

1. Introduction

In this paper we aim to develop a trust-region type method for solving the following bound-constrained minimization problem.

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad l \le x \le u, \end{aligned} \tag{1.1}$$

where $f : R^n \to R$ is continuously differentiable, $l \in (R \cup \{-\infty\})^n$, $u \in (R \cup \{\infty\})^n$, $l < u$. We denote the feasible set as $X = \{x \mid l \le x \le u\}$ and the strict interior feasible set as $X^0 = \{x \mid l < x < u\}$.

Trust-region methods for solving the bound-constrained minimization problem (1.1) have been studied extensively, (see [3]–[6]). We pay more attention to the trust-region methods proposed by Coleman and Li [3], Dennis and Vicente [6]. At kth iteration, by introducing a diagonal matrix, [3] presented a trust-region subproblem, which consisted of minimizing a

2001 Mathematics Subject Classification : 49M37, 90C20, 90C30.

1) The work was partly supported by the Scientific Research Foundation of Hunan Provincial Education Department, No. 02B021.

quadratic function subject only to an ellipsoidal constraint as follows:

$$\begin{aligned} \text{minimize} \quad & \varphi_k(s) = \nabla f_k^T s + \frac{1}{2} s^T (B_k + C_k) s \\ \text{subject to} \quad & \|D_k^{-1} s\| \le \Delta_k, \end{aligned} \tag{1.2}$$

where B_k, C_k, D_k are given special matrixes. Hence, the subproblem proposed in [3] possessed the form of an unconstrained trust-region subproblem and did not handle the bound-constraints explicitly. By using a step-back technique, Coleman and Li computed $\{x_k\}$ such that it satisfied strict feasibility. Elegant convergence results are obtained in [3]. Dennis and Vicente [6] considered another trust-region interior-point method for problem (1.1), which minimized the local quadratic function over a trust-region with the requirement that the iterative point had to be strictly feasible, i.e.,

$$\begin{aligned} \text{minimize} \quad & \varphi_k(s) \\ \text{subject to} \quad & \|S_k^{-1} s\| \le \Delta_k, \\ & \sigma_k(l - x_k) \le s \le \sigma_k(u - x_k), \end{aligned} \tag{1.3}$$

where S_k is given scale matrix and chosen $S_k = D_k$, $S_k = I_n$ in [6]. Nice convergence results are also obtained. Moreover, the idea of setting subproblem in [6] is extended to infinite-dimensional nonconvex minimization subject to pointwise bounds in [13] and to a class of nonlinear programming problems in [5].

The approach of the traditional trust-region method is similar. Namely, when the trial step is not accepted, we reduce the trust region radius and resolve the proposed subproblem. However, it is well-known that solving the trust-region subproblem is costly, which motivates us to study the trust-region method again. Unlike the existing trust-region methods, in this paper, a trust-region type method for solving problem (1.1) is presented. We adopt the subproblem version of Coleman and Li in [3], combine it with a line search technique. In particular, at each step, even if the trial step cannot be accepted, the solution of the subproblem (1.2) provides a descent direction of the objective function. Then we use a backtracking line-search to determine a steplength and get the next trial point. This combination of trust-region techniques and line-search techniques was introduced by Nocedal and Yuan in [11] for solving unconstrained optimization problems. We extend this technique to develop an iterative method for solving bound-constrained optimization problems. The advantage of the proposed method in this paper is twofold. First, it shares the advantages of trust-region methods. Second, at each step, the subproblem is solved only once. It is then reasonable to believe that the proposed method in this paper is cheaper than the existing trust-region methods. Under the same conditions of [3] and [6], we prove the convergence of the proposed algorithm. A numerical test shows that the new algorithm proposed in this paper is promising.

The paper is organized as follows. In Section 2 we give the preliminaries. The new algorithm is stated in Section 3. Global convergence of the new algorithm is proved in Section 4. In the last section, a numerical example is given.

Throughout this paper, the vector and matrix norms used are l_2 norm and subscripted indices k represents the evaluation of a function at kth step, for example, $f_k = f(x_k)$, etc.

2. Preliminaries

Denote $g(x) = \nabla f(x)$. The scaled matrix defined here is similar to the one of [3] and [6], i.e., a diagonal matrix whose diagonal elements are given by

$$(D(x))_{ii} = \begin{cases} (u_i - x_i)^{\frac{1}{2}}, & \text{if } g_i < 0 \text{ and } u_i < \infty, \\ (x_i - l_i)^{\frac{1}{2}}, & \text{if } g_i \geq 0 \text{ and } l_i > -\infty, \\ 1, & \text{if } g_i < 0 \text{ and } u_i = \infty, \\ 1, & \text{if } g_i \geq 0 \text{ and } l_i = -\infty. \end{cases} \tag{2.1}$$

Then from [3], [6] and [2], we have the following proposition.

Proposition 2.1 *$x^* \in X$ is a KKT point of (1.1) if and only if*

$$D^2(x^*)\nabla f(x^*) = 0. \tag{2.2}$$

Formula (2.2) provides the motivation for our algorithm. We transform the bound-constrained problem (1.1) to a problem of finding a local minimizer for some unconstrained problem and it shows that the sequence $\{x_k\}$ generated by our algorithm satisfies

$$\lim_{k\to\infty} \|D^2(x_k)\nabla f(x_k)\| = 0.$$

We note that the ith component of the function $D^2(x)$ is differentiable except at the point where $(g(x))_i = 0$. However, from the definition of

$D^2(x)$, this lack of smoothness is benign. Hence, we can define a *Jacobian* of $D^2(x)$ as follows:

$$(J_k)_{ii} \equiv (D^2(x))'_{ii} = \begin{cases} -1, & \text{if } (g(x))_i < 0, \\ 1, & \text{if } (g(x))_i > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

For more details, we prefer to see [3], [6] and [13]. Based on the Newton step for system (2.2), at kth iteration, we then set a trust-region subproblem of (1.1) as

$$\begin{cases} \text{minimize} & \psi_k(s) = g_k^T s + \dfrac{1}{2} s^T B_k s \\ \text{subject to} & \|D_k^{-1} s\| \leq \Delta_k, \end{cases} \tag{2.4}$$

where $B_k = H_k + C_k$, $C_k = D_k^{-1}\operatorname{diag}(g_k) J_k D_k^{-1}$, H_k is an approximation to $\nabla^2 f(x)$. Let $\hat{s} = D_k^{-1} s$, note that (2.4) is equivalent to the following subproblem:

$$\begin{cases} \text{minimize} & \hat{\psi}_k(\hat{s}) = \hat{g}_k^T \hat{s} + \dfrac{1}{2} \hat{s}^T \hat{B}_k \hat{s} = \psi_k(s) \\ \text{subject to} & \|\hat{s}\| \leq \Delta_k, \end{cases} \tag{2.5}$$

where $\hat{g}_k = D_k g_k$, $\hat{B}_k = D_k B_k D_k = D_k H_k D_k + \operatorname{diag}(g_k) J_k$. (2.5) is a standard trust-region subproblem of unconstrained optimization, we thus have the following important lemma.

Lemma 2.1 *If $\hat{s}_k$ is a solution of* (2.5), *then*

$$\hat{g}_k^T \hat{s}_k \leq -\frac{1}{2}\|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{2\|\hat{B}_k\|}\right\}. \tag{2.6}$$

Equivalently, the solution of (2.4) s_k *satisfies*

$$g_k^T s_k \leq -\frac{1}{2}\|D_k g_k\| \min\left\{\Delta_k, \frac{\|D_k g_k\|}{2\|D_k B_k D_k\|}\right\}. \tag{2.7}$$

The lemma above is a key result for us to construct new algorithm. Moreover, in order to ensure that all iterates are strictly feasible, we use a step-back technique, which is similar to the method proposed by Coleman and Li in [3]. When the trial step comes from (2.4) or (2.5), we solve the following problem to get a solution τ_k^*,

$$\min_{\tau\in[0,\min\{1,\alpha_k\}]} \phi(\tau) = \psi_k(\tau s_k). \tag{2.8}$$

where α_k expresses the stepsize along the direction d_k, i.e.,

$$\alpha_k = \min\left\{\max\left\{\frac{l_i-(x_k)_i}{(d_k)_i}, \frac{u_i-(x_k)_i}{(d_k)_i}\right\}, \ 1\le i\le n\right\}, \tag{2.9}$$

$$\begin{cases} \dfrac{l_i-(x_k)_i}{(d_k)_i} = \dfrac{u_i-(x_k)_i}{(d_k)_i} = +\infty, & \text{if } (d_k)_i = 0, \\ \alpha_k = +\infty, & \text{if } l = -\infty \text{ and } u = +\infty, \end{cases} \tag{2.10}$$

where $(d_k)_i$ and $(x_k)_i$ express the ith component of d_k and x_k. From (2.8)–(2.10) we can easily prove that $x_k + \tau_k^* s_k \in X$. Finally we use step-back method to choose θ_k such that

$$\theta_k \in [\theta_l, 1], \quad \theta_k - 1 = O(\|s_k\|), \quad x_k + \theta_k\tau_k^* s_k \in X^0, \tag{2.11}$$

where $\theta_l > 0$ is a constant. We denote $\psi_k^*(s_k) = \psi_k(\theta_k\tau_k^* s_k)$.

On the other hand, to analyze the convergence of the trust-region method, a sufficient reduction of the quadratic model $\psi_k(s)$ is required. Here we consider the scaled gradient direction for subproblem of (2.4), which is considered by many researchers (for example, [3], [5], [6], [10]) and often called "Cauchy step" associated with the trust-region subproblem (2.4). In addition, we also require strict feasibility for the point generated by the scaled gradient direction and we deal with the problem by using the same technique mentioned above. The following problem is solved first,

$$\min_{\tau\in[0,\min\{\Delta_k,\alpha_k\}]} \quad \psi_k(\tau p_k), \tag{2.12}$$

where $p_k = -D_k\frac{\hat{g}_k}{\|\hat{g}_k\|} \in \mathrm{Span}\{-D_k^2 g_k\}$. Denote the solution of (2.12) as $\bar{\tau}_k^*$. Then choose $\bar{\theta}_k$ such that

$$\bar{\theta}_k \in [\theta_l, 1], \quad \bar{\theta}_k - 1 = O(\|p_k\|), \quad x_k + \bar{\theta}_k\bar{\tau}_k^* p_k \in X^0. \tag{2.13}$$

We denote $\psi_k^*(-D_k^2 g_k) = \psi_k(\bar{\theta}_k\bar{\tau}_k^* p_k)$.

3. Combining algorithm

In this section, we give the steps of the new algorithm. The idea of our algorithm comes from Lemma 2.1, which implies that any solution of the subproblem (2.4) provides a descent direction of objective function at x_k. Moreover, the step-back technique never affects the descent. Therefore, if the trial step is accepted, we can obtain the next iteration. Otherwise, we can use Armijo line search to find the next iterative point. We state our algorithm as follows:

Algorithm 3.1

Step 0. Given $x_0 \in X^0$, $\beta \in (0,1)$, $\mu \in (0,1/2)$, $0 < \eta_1 < \eta_2 \leq 1$, $\Delta_{\max} \geq \Delta_0 \geq \Delta_{\min} > 0$, $0 < r_1 < 1 \leq r_2$, the symmetric matrix $H_0 \in R^{n\times n}$, $\epsilon > 0$, $k := 0$.

Step 1. Compute g_k, B_k, D_k. If $\|D_k g_k\| \leq \epsilon$ stop and output x_k.

Step 2. Solve subproblem (2.5) and (2.8) to obtain s_k and τ_k^*. Choose θ_k to satisfy (2.11). Let $d_k = \theta_k \tau_k^* s_k$. Solve (2.12) to obtain $\bar{\tau}_k^*$ and choose $\bar{\theta}_k$ to satisfy (2.13), let $\bar{d}_k = \bar{\theta}_k \bar{\tau}_k^* p_k$.

Step 3. If $\psi_k^*(s_k) > \psi_k^*(-D_k g_k)$, set $d_k = \bar{d}_k$.

Step 4. Compute $\rho_k = \frac{f(x_k) - f(x_k+d_k) - \frac{1}{2} d_k^T C_k d_k}{-\psi_k(d_k)}$. If

$$\rho_k \geq \eta_1, \tag{3.1}$$

set $x_{k+1} = x_k + d_k$ and goto Step 6.

Step 5. (Armijo line-search) Find the minimum positive integer i_k such that

$$f(x_k) - f(x_k + \beta^i d_k) \geq -\mu \beta^i d_k^T \nabla f(x_k). \tag{3.2}$$

Set $x_{k+1} = x_k + \beta^{i_k} d_k$, $\Delta_{k+1} \in [\|x_{k+1} - x_k\|, r_1 \Delta_k]$ and goto Step 7.

Step 6. If $\rho_k < \eta_2$, set $\Delta_{k+1} \in [r_1 \Delta_k, \Delta_k]$. Otherwise $\Delta_{k+1} \in \min\{r_2 \Delta_k, \Delta_{\max}\}$.

Step 7. Update H_k as H_{k+1}, $k := k+1$ goto Step 1.

Remarks

1. We do not need to solve the subproblem (2.5) exactly and only need that s_k satisfies (2.6). The paper [11] gave an algorithm to compute this approximate solution.

2. From step 3 of Algorithm 3.1 , for all k we have

$$\frac{\psi_k(d_k)}{\psi_k^*(-D_k^2 g_k)} \geq 1 \geq \beta. \tag{3.3}$$

3. The main difference between the Algorithm 3.1 with [3], [6] and other traditionary trust-region methods is that we need not to solve the subproblem (2.4) or (2.5) repeatedly when (3.1) does not hold.

4. Global convergence for Algorithm 3.1

First we give general global assumptions as follows:

AS.1 $f \in C^2(X)$.

AS.2 For $x_0 \in X^0$, $L = \{x \mid x \in X,\ f(x) \leq f(x_0)\}$ is compact.

AS.3 For all k, there exists a constant $M_H > 0$ such that $\|H_k\| \leq M_H$.

AS.4 there exists a constant $M_g > 0$ such that for all $x \in L$, $\|g(x)\| \leq M_g$.

From the above assumptions, there exist constants $M_D > 0$, $M_B > 0$ such that for all k we have

$$\|D_k\| \leq M_D, \quad \|\hat{B}_k\| \leq M_B. \tag{4.1}$$

From now on, we always suppose that the above assumptions hold. Denote an index set as

$$K = \{k \mid x_{k+1} = x_k + \beta^{i_k} d_k\}, \tag{4.2}$$

which expresses the index of using line search method.

The next lemma shows that Algorithm 3.1 is well-defined.

Lemma 4.1 *There exists a minimum positive integer i_k such that* (3.2) *holds.*

Proof. From Algorithm 3.1, the trial step is chosen as $d_k = \theta_k \tau_k^* s_k$ or $d_k = \bar{\theta}_k \bar{\tau}_k^* p_k$. Since the algorithm does not stop, we know that $\|D_k g_k\| > \epsilon$. Consider two cases:

Case I: $d_k = \theta_k \tau_k^* s_k$. From Lemma 2.1 we have

$$-d_k^T \nabla f_k \geq \frac{1}{2}\theta_k \tau_k^* \|D_k \nabla f_k\| \min\left\{\Delta_k, \frac{\|D_k \nabla f_k\|}{2\|\hat{B}_k\|}\right\}.$$

which, combined (4.1) with $\|D_k \nabla f_k\| \geq \epsilon$, yields

$$-d_k^T \nabla f_k \geq \frac{1}{2}\theta_k \tau_k^* \epsilon \min\left\{\Delta_k, \frac{\epsilon}{2M_B}\right\} > 0. \tag{4.3}$$

Then, from directional derivative arguments, we know that there exists i_k such that (3.2) holds.

Case II: $d_k = \bar{\theta}_k \bar{\tau}_k^* p_k = -\bar{\theta}_k \bar{\tau}_k^* \frac{D_k^2 g_k}{\|D_k g_k\|}$, we have

$$-d_k^T \nabla f_k = \bar{\theta}_k \bar{\tau}_k^* \|D_k \nabla f_k\| \geq \bar{\theta}_k \bar{\tau}_k^* \epsilon > 0. \tag{4.4}$$

Similar to the case I, there exists i_k such that (3.2) holds. $\square$

The following lemma is proved by Coleman and Li (see Lemma 3.1 in [3]), and gives the reduction of the quadratic model which plays an important role in convergence of trust-region method.

Lemma 4.2 *Assume that d_k is computed by Algorithm* 3.1. *Then,*

$$-\psi_k(d_k) \geq \frac{1}{2}\|\hat{g}_k\| \min\left\{ \Delta_k, \frac{\|\hat{g}_k\|}{\|\hat{B}_k\|}, \frac{\|\hat{g}_k\|}{\|g_k\|_\infty} \right\}. \tag{4.5}$$

From the global assumptions, we get the following lemma.

Lemma 4.3 *There exists a constant $M > 0$ such that for all k we have*

$$\left| f(x_k) - f(x_k + d_k) - \frac{1}{2} d_k^T C_k d_k - (-\psi_k(d_k)) \right| \leq M \|d_k\|^2. \tag{4.6}$$

The next two lemmas can show some properties of the sequence.

Lemma 4.4 *Let K be defined by* (4.2). *If there is a subset $K_1 \subset K$ such that for all $k \in K_1$, $\|D_k g_k\| > \epsilon$, then there exist $\Delta^* > 0$ and $\tau^* > 0$ such that for all $k \in K_1$ we have*

$$\Delta_k \geq \Delta^*, \tag{4.7}$$

$$\min\{\tau_k^*, \bar{\tau}_k^*\} > \tau^*. \tag{4.8}$$

Proof. From the definition of K we know that the trial step is not accepted, i.e.,

$$\left| \frac{f(x_k) - f(x_k + d_k) - \frac{1}{2} d_k^T C_k d_k}{-\psi_k(d_k)} - 1 \right| > 1 - \eta_1. \tag{4.9}$$

For the two cases ($d_k = \theta_k \tau_k^* s_k$ and $d_k = \bar{\theta}_k \bar{\tau}_k^* p_k$), we have $\|d_k\| \leq M_D \Delta_k$, which combines (4.9) with Lemma 4.2 and Lemma 4.3 to yield

$$1 - \eta_1 < \frac{M M_D^2 \Delta_k}{\frac{1}{2}\epsilon \min\left\{1, \frac{\epsilon}{M_B \Delta_{\max}}, \frac{\epsilon}{M_g \Delta_{\max}}\right\}}.$$

This implies that

$$\Delta_k > \frac{(1-\eta_1)\frac{1}{2}\epsilon \min\{1, \frac{\epsilon}{M_B\Delta_{\max}}, \frac{\epsilon}{M_g\Delta_{\max}}\}}{MM_D^2} \equiv \Delta^*.$$

So (4.7) is proved. Now we prove (4.8). From Lemma 4.2 and (4.7) we have

$$\frac{1}{2}\epsilon \min\left\{1, \frac{\epsilon}{M_B\Delta_{\max}}, \frac{\epsilon}{M_g\Delta_{\max}}\right\}\Delta^* \le -\psi_k(d_k). \tag{4.10}$$

On the other hand, we have two choices for d_k in Algorithm 3.1. For $d_k = \theta_k\tau_k^* s_k$ we have

$$\begin{aligned}
-\psi_k(d_k) &= -\theta_k\tau_k^* g_k^T s_k - \frac{1}{2}\theta_k^2(\tau_k^*)^2 \hat{s}_k^T \hat{B}_k \hat{s}_k \\
&\le \tau_k^* M_g M_D \Delta_{\max} + \frac{1}{2}(\tau_k^*)^2 M_B \Delta_{\max}^2 \\
&\le \left(M_g M_D \Delta_{\max} + \frac{1}{2} M_B \Delta_{\max}^2\right)\tau_k^*,
\end{aligned}$$

which combines with (4.10) to yield

$$\tau_k^* \ge \frac{\frac{1}{2}\epsilon \min\{1, \frac{\epsilon}{M_B\Delta_{\max}}, \frac{\epsilon}{M_g\Delta_{\max}}\}}{M_g M_D \Delta_{\max} + \frac{1}{2} M_B \Delta_{\max}^2} \equiv \tau_1^*. \tag{4.11}$$

For $d_k = \bar{\theta}_k \bar{\tau}_k^* p_k$, $p_k = -D_k \frac{\hat{g}_k}{\|\hat{g}_k\|}$, we deduce that

$$\begin{aligned}
-\psi_k(d_k) &= \bar{\theta}_k \bar{\tau}_k^* \|g_k\| - \frac{1}{2}\bar{\theta}_k^2(\bar{\tau}_k^*)^2 \frac{\hat{g}_k^T \hat{B}_k \hat{g}_k}{\|\hat{g}_k\|^2} \\
&\le \left(M_D M_g + \frac{1}{2} M_B\right)\bar{\tau}_k^*.
\end{aligned}$$

From (4.10) again we have

$$\bar{\tau}_k^* \ge \frac{\frac{1}{2}\epsilon \min\{1, \frac{\epsilon}{M_B\Delta_{\max}}, \frac{\epsilon}{M_g\Delta_{\max}}\}\Delta^*}{M_D M_g + \frac{1}{2} M_B} \equiv \tau_2^*. \tag{4.12}$$

Denote $\tau^* = \min\{\tau_1^*, \tau_2^*\} > 0$. Then (4.8) follows. □

Lemma 4.5 *Under the conditions of Lemma* 4.4, *there exist constants* $\delta^* > 0$ *and* $\beta^* > 0$ *such that for all* $k \in K_1$

$$d_k^T \nabla f_k \le -\delta^* < 0, \tag{4.13}$$

$$\beta^{i_k} \geq \beta^* > 0. \tag{4.14}$$

Proof. From (4.3), (4.4) in the proof of Lemma 4.1, the choice of θ_k and $\bar{\theta}_k$ we have

$$-d_k^T \nabla f_k \geq \theta_l \epsilon \min\left\{\frac{1}{2}\tau_k^* \min\left\{\Delta_k, \frac{\epsilon}{2M_B}\right\}, \bar{\tau}_k^*\right\}. \tag{4.15}$$

Then (4.13) follows from (4.7) and (4.8).

From the global assumptions and $\|d_k\| \leq M_D \Delta_k \leq M_D \Delta_{\max}$ we have:

$$\begin{aligned} & f(x_k) - f(x_k + \beta^i d_k) \\ & = -\beta^i d_k^T \nabla f_k - (\beta^i)^2 d_k^T \nabla^2 f(\xi_k) d_k \\ & \geq -\beta^i d_k^T \nabla f_k - M_f M_D^2 \Delta_{\max}^2 (\beta^i)^2 \\ & = -\mu \beta^i d_k^T \nabla f_k - (1-\mu)\beta^i d_k^T \nabla f_k - M_f M_D^2 \Delta_{\max}^2 (\beta^i)^2 \end{aligned} \tag{4.16}$$

where $\xi_k \in (x_k, x_k + \beta^i d_k)$. Hence, if β^i satisfies that

$$(1-\mu)\beta^i(-d_k^T \nabla f_k) \geq M_f M_D^2 \Delta_{\max}^2 (\beta^i)^2 \tag{4.17}$$

(3.2) holds in Algorithm 3.1. Obviously, if $\beta^i < \frac{(1-\mu)\delta^*}{M_f M_D^2 \Delta_{\max}^2}$, (4.17) holds since (4.13). On the other hand, From the definition of i_k, the following statement holds.

$$\beta^{i_k} \geq \frac{\beta(1-\mu)\delta^*}{M_f M_D^2 \Delta_{\max}^2} \equiv \beta^*.$$

Therefore (4.14) is proved. □

Next we state the main convergence results of Algorithm 3.1.

Theorem 4.1 *Let $\{x_k\}$ be generated by Algorithm 3.1. Under the global assumptions, we have*

$$\liminf_k \|D_k g_k\| = 0. \tag{4.18}$$

Proof. We consider the two cases to prove the theorem: K is both finite and infinite.

Case I: K is finite. We know that, in this case, there exists a positive integer $\bar{k}$ independent of k such that for all $k > \bar{k}$, each trial step can be accepted by the trust region method. Then our algorithm reduces to the algorithm in [3]. The conclusion of (4.18) follows from the Theorem 3.4 in [3].

Case II: K is infinite. We first prove the following result.

$$\lim_{k\in K}\|D_k g_k\| = 0. \tag{4.19}$$

The statement (4.19) is proved by contradiction. Assume that there exist an infinite set $K_1 \subset K$ and $\epsilon > 0$ such that $\|D_k g_k\| \geq \epsilon$ for all $k \in K_1$. From Lemma 4.5 and Algorithm 3.1 we have, for $k \in K_1$,

$$f(x_k) - f(x_{k+1}) \geq -\mu\beta^{i_k} d_k^T \nabla f_k \geq \mu\beta^* \delta^*.$$

Since $\{f(x_k)\}$ is monotonically decreasing and bounded below, we deduce:

$$\begin{aligned}\infty > \sum_{k=0}^{\infty}(f(x_k) - f(x_{k+1})) &\geq \sum_{k\in K}(f(x_k) - f(x_{k+1}))\\ \geq \sum_{k\in K_1}(f(x_k) - f(x_{k+1})) &\geq \sum_{k\in K_1}\mu\beta^*\delta^* = \infty,\end{aligned}$$

which is a contradiction. So (4.19) holds. (4.18) directly follows (4.19).

□

Theorem 4.2 *Let* $\{x_k\}$ *be generated by Algorithm* 3.1. *Under the global assumptions, the following statement holds*:

$$\lim_{k\to\infty}\|D_k g_k\| = 0. \tag{4.20}$$

Proof. The proof is given by contradiction. Assume that there exist an infinite sequence $\{m_i\}$ and a constant $\epsilon_1 \in (0, 1)$ such that for all $k \in \{m_i\}$

$$\|D_{m_i} g_{m_i}\| > \epsilon_1. \tag{4.21}$$

On the other hand, from Theorem 4.1, for any $\epsilon_2 \in (0, \epsilon_1)$, there exists a subsequence of $\{l_i\}$ (also called, without loss of generality, $\{l_i\}$) such that

$$\|D_k g_k\| \geq \epsilon_2, \quad m_i \leq k < l_i, \quad \|D_{l_i} g_{l_i}\| < \epsilon_2. \tag{4.22}$$

Let us consider the kth iteration. If the trial step is accepted by the trust-region method, i.e., $k \notin K$, from (3.1) in Algorithm 3.1 and Lemma 4.2 we have

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \frac{1}{2}\eta_1\epsilon_2 \min\left\{1, \frac{\epsilon_2}{M_B\Delta_{\max}}, \frac{\epsilon_2}{M_g\Delta_{\max}}\right\}\Delta_k \\ &\equiv \kappa_1\epsilon_2\Delta_k, \end{aligned} \tag{4.23}$$

where $\kappa_1 = \frac{1}{2}\eta_1 \min\{1, \frac{\epsilon_2}{M_B\Delta_{\max}}, \frac{\epsilon_2}{M_g\Delta_{\max}}\} > 0$. If $k \in K$, from (3.2) in Algorithm 3.1, (4.15), Lemma 4.4 and Lemma 4.5 we get

$$\begin{aligned} &f(x_k) - f(x_{k+1}) \\ &\geq -\mu\beta^{i_k} d_k^T \nabla f(x_k) \\ &\geq \mu\beta^*\theta_l\epsilon_2\tau^* \min\left\{\frac{1}{2}\min\{1, \frac{\epsilon_2}{2M_B\Delta_{\max}}\}, \frac{1}{\Delta_{\max}}\right\}\Delta_k \\ &\equiv \kappa_2\epsilon_2\Delta_k, \end{aligned} \tag{4.24}$$

where $\kappa_2 = \mu\beta^*\theta_l\tau^* \min\{\frac{1}{2}\min\{1, \frac{\epsilon_2}{2M_B\Delta_{\max}}\}, \frac{1}{\Delta_{\max}}\} > 0$. Hence, from (4.23) and (4.24), if each k satisfied (4.22) we have

$$f(x_k) - f(x_{k+1}) \geq \min\{\kappa_1, \kappa_2\}\epsilon_2\Delta_k. \tag{4.25}$$

On the other hand, from Algorithm 3.1 and the assumptions we know that there exists a constant $\kappa_3 > 0$, for all k, $\|x_k - x_{k+1}\| \leq \kappa_3\Delta_k$, which combining with (4.25) to yield

$$f(x_k) - f(x_{k+1}) \geq \epsilon_3\|x_{k+1} - x_k\|, \tag{4.26}$$

where $\epsilon_3 = \epsilon_2\frac{\min\{\kappa_1,\kappa_2\}}{\kappa_3}$. The next proof follows the same steps as the proof of Theorem 3.5 in [3] and the proof of convergence in [10]. So the theorem is proved. □

5. Numerical example

In this section we show a numerical example to illustrate the advantage of Algorithm 3.1. Two algorithms are considered. One is the pure trust-region algorithm, denoted by PTR, which uses the traditional trust-region method, i.e., if $\rho_k < \eta_1$ in Algorithm 3.1, we then reduce the trust-region radius $\Delta_k = 0.5\Delta_k$ and goto step 2 to resolve the trust-region subproblem (2.4) or (2.5). Another algorithm is Algorithm 3.1, i.e., combining trust-region method and line search method, denoted by CTL. The numerical example comes from [8] problem 38.

$$\begin{aligned} \text{minimize} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & \qquad + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \\ \text{subject to} \quad & \qquad -10 \le x_i \le 10, \quad (i = 1, 2, \ldots, 4) \end{aligned}$$

The optimal point of this example is $x^* = (1, 1, 1, 1)$.

A MATLAB subroutine has been coded. The constants for the two algorithms are chosen as follows:

$$\beta = 0.5,\ \mu = 0.4,\ \eta_1 = 0.25,\ \eta_2 = 0.75,\ \Delta_{\max} = 100,$$
$$\Delta_{\min} = 1.0e - 4,\ \Delta_0 = 3,\ r_1 = 0.5,\ r_2 = 2.$$

The stop condition is $\epsilon = 1.0e - 5$. The subproblem is solved in truncated conjugate gradient method proposed by Yuan [14]. The computing results are reported in Table 5.1. The CPU (sec.) time is only used to compare two algorithms.

Table 5.1.

start point	CTL algorithm			PTR algorithm		
x_0	k	k_s	CPU	k	k_s	CPU
$[0,0,0,0]$	60	60	0.66	81	89	0.94
$[-1,-1,-1,-1]$	259	259	2.91	212	341	3.57
$[5,5,5,5]$	76	76	0.71	76	76	0.76
$[2,8,2,8]$	26	26	0.27	105	108	1.10
$[-1,9,9,9]$	164	164	1.71	160	203	2.09
$[-1,-1,0,0]$	143	143	1.82	194	251	2.58
$[8,8,8,8]$	199	199	1.60	199	199	1.60
$[6,0,6,0]$	38	38	0.49	38	38	0.49

where k denotes the iterative numbers, k_s expresses the total iterative numbers for solving the trust-region subproblem (2.4), CPU denotes the time taken for solving the example from different start point. From Table 5.1 we can see that, in most cases, the iterative number of CTL algorithm is less than PTR algorithm. In some cases, though the main iterative number of CTL algorithm is more than PTR, the time of solving subproblem and the time taken of CPU are less than PTR algorithm. This shows Algorithm 3.1 is effective.

References

[1] Burke J.V., *A robust trust-region method for constrained nonlinear programming problem*. SIAM J. Optim. **2** (1992), 325–347.

[2] Coleman T.F. and Li Y., *On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds*. Math. Prog. **67** (1994), 189–224.

[3] Coleman T.F. and Li Y., *An interior trust-region approach for nonlinear minimization subject to bounds*. SIAM J.Optim. **6** (1996), 418–445.

[4] Conn A.R., Gould N. and Toint Ph.L., *Global convergence of a class of trust-region algorithm for optimization with simple bounds*. SIAM J.Numer. Anal. **25** (1988), 433–460.

[5] Dennis J.E., Heinkenschloss M. and Vicente L.N., *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*. SIAM J. Control and Optimization **36** (1998), 1750–1794.

[6] Dennis J.E. and Vicente L.N., *Trust region interior-point algorithms for minimization methods with simple bounds*. In H. Fischer, B. Riedmuller, and S. Schaffler, editors, Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter, Heidelberg, (1996) 97–107.

[7] Friedlander A., Martinez J.M. and Santos S.A., *A new trust-region algorithm for bound constrained minimization*. Appl. Math. Optim. **30** (1994), 235–266.

[8] Hock W. and Schittkowski K., *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Economics and Mathematical Systems, No. **187**, Springer-Verlag, Berlin Heidelkery, New York, (1981).

[9] Liang X.M. and Xu C.X., *A trust-region algorithm for bound constrained minimization*. Optimization **41** (1997), 279–289.

[10] Moré J.J., *Recent development in algorithms and software for trust region methods*. In: G.A.Watson, eds., Numerical Analysis Dundee 1977, Lecture Notes in Mathematics 630, Springer-Verlag, Berlin, (1983), 258–287.

[11] Nocedal J. and Yuan Y., *Combining trust region and line search techniques*. Technical Report OTC 98/04, Optimization Technology Center, (1998).

[12] Tong X. and Zhou S., *Global convergence of a trust-region algorithm for solving inequality constrained optimization*. Hokkaido Mathematical Journal, Vol. xxx, No.1 (2001), 113-136.

[13] Ulbrich M., Ulbrich S. and Heinkenschloss M., *Global convergence of trust-region interior-point algorithms for infinite-dimensional nonconvex minimization subject to pointwise bounds*. SIAM J. Control and Optimization **37** (1999), 731–764.

[14] Yuan Y., *On the truncated conjugate gradient method*. Math. Prog.**87** (2000), 561–573.

Xiaojiao Tong
Department of Mathematics and Computer
Changsha University of Electric Power
Changsha 410077
P. R. China
E-mail: luoket@cs.hn.cn

Shuzi Zhou
Department of Mathematics
Hunan University
Changsha 410082
P. R. China