Journal of Computational Mathematics, Vol.22, No.6, 2004, 865–872.

A LINE SEARCH AND TRUST REGION ALGORITHM WITH TRUST REGION RADIUS CONVERGING TO ZERO [*1)]

Jin-yan Fan
(*Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, China*)

Wen-bao Ai
(*School of Sciences, Beijing University of Posts and Telecommunications, Beijing 100876, China*)

Qun-ying Zhang
(*Department of Mathematics, Yangzhou University, Yangzhou 225009, China*)

Abstract

In this paper, we present a new line search and trust region algorithm for unconstrained optimization problem with the trust region radius converging to zero. The new trust region algorithm performs a backtracking line search from the failed point instead of resolving the subproblem when the trial step results in an increase in the objective function. We show that the algorithm preserves the convergence properties of the traditional trust region algorithms. Numerical results are also given.

Mathematics subject classification: 90C30, 65K05.
Key words: Unconstrained optimization, Trust region, Line search.

1. Introduction

In this paper, we consider the line search and trust region method for the unconstrained optimization problem

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

where $f : R^n \to R$ is continuously differentiable. In every iteration, a trial step is computed by solving the subproblem

$$\begin{array}{ll} \min\limits_{d\in R^n} & g_k^T d + \frac{1}{2} d^T B_k d := \phi_k(d) \\ s.\ t. & ||d|| \le \Delta_k, \end{array} \tag{1.2}$$

where $g_k = \nabla f(x_k), B_k$ is a $n \times n$ symmetric matrix which approximates the Hessian of f at x_k, $\Delta_k > 0$ is the current trust region radius, and $||\cdot||$ refers to the 2-norm.

Let d_k be the solution of (1.2). The predicted reduction is defined by the reduction of the approximate model, that is,

$$Pred_k = \phi_k(0) - \phi_k(d_k), \tag{1.3}$$

and the actual reduction is defined by

$$Ared_k = f(x_k) - f(x_k + d_k). \tag{1.4}$$

The ratio between these two reductions is defined by

$$r_k = \frac{Ared_k}{Pred_k}, \tag{1.5}$$

[*] Received November 19, 2002.
[1)] This work is partially supported Chinese NSFC grants 10371076, Research Grands for Young Teachers of Shanghai Jiaotong University, and Institute of Computational Science, Shanghai University.

and it plays a key role in the traditional trust region algorithm (TTR) to decide whether the trial step is acceptable and to adjust the new trust region radius. If the trial step is not successful, then we reject it, reduce the trust region radius, and resolves the subproblem (1.2), otherwise, we accept the trial step, and enlarge the trust region radius. That is, in TTR, we choose:

$$x_{k+1} = \begin{cases} x_k + d_k, & \text{if } r_k > c_0, \\ x_k, & \text{otherwise}, \end{cases} \tag{1.6}$$

where $c_0 \in [0, 1)$ is a small constant, and choose

$$\Delta_{k+1} \in \begin{cases} [c_3||d_k||,\ c_4\Delta_k], & \text{if } r_k < c_2, \\ [\Delta_k,\ c_1\Delta_k] & \text{otherwise}, \end{cases} \tag{1.7}$$

where $0 < c_3 < c_4 < 1 < c_1, 0 \le c_0 \le c_2 < 1$ are constants.

In TTR, when the sequence $\{x_k\}$ converges to the minimizer x^* of the objective function f, the ratio of the actual reduction and the predicted reduction r_k will converge to 1. For the details of trust region algorithms, please see [8, 9, 10]. It then follows from the updating rule of the trust region radius (1.7) that Δ_k will be larger than a positive constant for all sufficiently large k, hence, the trust region will not play the role at the end. In fact, it suffices for the convergence that Δ_k be larger than $O(||x_k - x^*||)$ at every iteration. To prevent the trial step from being too large near the minimizer, we present a trust region algorithm for (1.1) with the trust region radius converging to zero [1]. We choose

$$\Delta_k = \mu_k ||g_k||, \tag{1.8}$$

where μ_k is updated according to the ratio r_k.

As we know, to resolve the subproblem (1.2) can be costly, since this requires solving one or more linear systems as follows:

$$(B_k + \lambda I)d = -g_k, \tag{1.9}$$

while line search methods require little computation to decide a new point. Nocedal and Yuan creatively combine the trust region technique and line search technique in [5]. In this paper, we apply the line search technique to our trust region algorithm with the trust region converging to zero. The subproblem is solved by the algorithm given by Nocedal & Yuan in [5], hence the trial step d_k is always a direction of sufficient descent for the objective function. Thus we do not need to resolve the subproblem (1.2) when $f(x_k + d_k) > f(x_k)$, in stead we can carry out the backtracking line search along d_k until we obtain the new trial point at which the value of the objective function is less than $f(x_k)$.

In the next section, we present the new line search and trust region algorithm with the trust region converging to zero, and show that the new algorithm preserves the global convergence of the traditional trust region algorithm. In section 3, we discuss the superlinear convergence of the algorithm. Finally in section 4, some numerical results are given.

2. The Algorithm and Global Convergence

In this section, we first give some properties of the trust region subproblem (1.2), then present our new line search and trust region algorithm with the trust region converging to zero, finally we discuss the global convergence of the new algorithm.

The following results are well known(see Moré and Sorensen [4] and Gay [2]).

Lemma 2.1. *A vector $d^* \in R^n$ is a solution of the problem*

$$\begin{array}{ll} \min\limits_{d \in R^n} & g^T d + \frac{1}{2} d^T B d := \phi(d) \\ s.t. & ||d|| \le \Delta, \end{array} \tag{2.1}$$

where $g \in R^n, B \in R^{n\times n}$ is a symmetric matrix and $\Delta > 0$, if and only if $||d^*|| \le \Delta$ and there exists $\lambda^* \ge 0$ such that

$$(B + \lambda^* I)d^* = -g, \tag{2.2}$$

$$(B + \lambda^* I) \ge 0, \tag{2.3}$$

and

$$\lambda^*(\Delta - ||d^*||) = 0. \tag{2.4}$$

Lemma 2.2. *(Powell 1975) If d^* is a solution of (2.1), then*

$$\phi(0) - \phi(d^*) \ge \frac{1}{2}||g|| \min\{\Delta, ||g||/||B||\}. \tag{2.5}$$

The vector d_k can be calculated by dog-leg type techniques (see Powell [6]) or by applying the Newton's method to the following nonlinear equation (see Gay [2], Moré and Sorensen[4]),

$$\phi(\lambda) = \frac{1}{||(B_k + \lambda I)^{-1} g_k||} - \frac{1}{\Delta_k} = 0. \tag{2.6}$$

The subproblem can also be solved by a preconditioned conjugate gradient method (Steihaug [7]). In fact, the global convergence only requires that the predicted reduction satisfies

$$\phi_k(0) - \phi_k(d_k) \ge \eta||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}, \tag{2.7}$$

where η is a positive constant. To compute a vector d_k that satisfies (2.7) is usually much easier than solving (1.2) exactly. Recently, Nocedal and Yuan [5] propose a novel algorithm that solves (2.1) approximately. The solution d^* can be sufficiently downhill which is desirable for line search. We present the algorithm as follows:

Algorithm 2.1. *Step 1 Given constants $\gamma > 1, \epsilon > 0$, set $\lambda := 0$; if B is positive definite, go to Step 2; else find $\lambda \in [0, ||B|| + (1+\epsilon)||g||/\Delta]$ such that $B + \lambda I$ is positive definite.*

Step 2 Factorize $B + \lambda I = R^T R$, where R is upper triangular; solve $R^T R d = -g$ for d.

Step 3 If $||d|| \le \Delta$, stop; else solve $R^T q = d$ for q; and compute

$$\lambda := \lambda + \frac{||d||^2}{||q||^2} \frac{\gamma||d|| - \Delta}{\Delta}; \tag{2.8}$$

go to Step 2.

Nocedal and Yuan show that if d_k is computed by Algorithm 2.1, then there exists a positive constant τ such that

$$\phi_k(0) - \phi_k(d_k) \ge \tau||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}, \tag{2.9}$$

and

$$d_k^T g_k \le -\tau||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}. \tag{2.10}$$

Thus d_k is a direction of sufficient descent in the sense that the angle between d_k and $-g_k$ will be bounded away from $\pi/2$ if $||g_k||$ is bounded away from zero and $||B_k||$ is bounded above. Hence, if d_k is not acceptable, we can safely perform a backtracking line search along d_k: we find the minimum positive integer i_k such that

$$f(x_k + \alpha^{i_k} d_k) < f(x_k), \tag{2.11}$$

where $\alpha \in (0,\ 1)$ is a positive constant. We now present our new algorithm as follows:

Algorithm 2.2. *Step 1* *Given* $x_1 \in R^n, B_1 \in R^{n\times n}$ *symmetric*, $\varepsilon \geq 0, c_2 > c_0 \geq 0, 0 < c_5 < 1 < c_6, 0 < c_7, c_8 < 1, 0 < \alpha < 1, \mu_1 > 0$, *set* $\Delta_1 = \mu_1 ||g_1||, k := 1$.

Step 2 *If* $||g_k|| \leq \varepsilon$, *then stop; else solve (1.2) so that* $||d_k|| \leq \Delta_k$ *and (2.9)–(2.10) are satisfied. (This can be done by Algorithm 2.1).*

Step 3 *Compute* $f(x_k + d_k)$; *if* $f(x_k + d_k) < f(x_k)$, *go to Step 4; else find the minimum positive integer* i_k *such that*

$$d_k^{i_k} = \alpha^{i_k} d_k \qquad \text{and} \qquad f(x_k + d_k^{i_k}) < f(x_k); \tag{2.12}$$

compute

$$x_{k+1} = x_k + d_k^{i_k}, \tag{2.13}$$

$$\mu_{k+1} = c_7 \mu_k; \tag{2.14}$$

go to Step 5.

Step 4 *Compute* $r_k = Ared_k / Pred_k$ *and*

$$x_{k+1} = x_k + d_k; \tag{2.15}$$

choose μ_{k+1} *as*

$$\mu_{k+1} = \begin{cases} c_5 \mu_k, & \text{if } r_k < c_2, \\ c_6 \mu_k, & \text{if } r_k \geq c_2 \text{ and } ||d_k|| > c_8 \Delta_k, \\ \mu_k, & \text{otherwise.} \end{cases} \tag{2.16}$$

Step 5 *Compute*

$$\Delta_{k+1} = \mu_{k+1} ||g_{k+1}||; \tag{2.17}$$

update B_{k+1}; $k := k + 1$; *go to Step 2.*

Theorem 2.1. *Assume that* $f(x)$ *is differentiable and bounded below,* $g(x)$ *is uniformly continuously. If there exists a positive constant* M *such that*

$$||B_k|| \leq M \qquad \text{and} \qquad ||g_k|| \leq M \tag{2.18}$$

hold for all k, *then it follows that*

$$\liminf_{k\to\infty} ||g_k|| = 0. \tag{2.19}$$

Proof. If the theorem is not true, then there exists a positive constant δ such that

$$||g_k|| \geq \delta \tag{2.20}$$

holds for all k. Define the set

$$\mathcal{K} = \{k \mid r_k \geq c_2\}. \tag{2.21}$$

Since $f(x)$ is bounded below, we have from (2.18),(2.20) and (2.9) that

$$\begin{aligned} +\infty &> \sum_{i=1}^{\infty} (f_k - f_{k+1}) \\ &\geq \sum_{k\in\mathcal{K}} c_2(\phi_k(0) - \phi_k(d_k)) \\ &\geq \sum_{k\in\mathcal{K}} \delta\tau c_2 \min\{\Delta_k, \frac{\delta}{M}\}. \end{aligned} \tag{2.22}$$

The above relation and (2.17) indicate that

$$\sum_{k\in\mathcal{K}} \min\{\mu_k, \frac{1}{M}\} < +\infty. \tag{2.23}$$

If $\mathcal{K}$ is finite, we have $\mu_{k+1} = c_5\mu_k$ or $\mu_{k+1} = c_7\mu_k$ for all sufficiently large k, thus it follows

$$\lim_{k\to\infty} \mu_k = 0. \tag{2.24}$$

If $\mathcal{K}$ is infinite, then we have from (2.23) that

$$\lim_{k\in\mathcal{K}} \mu_k = 0, \tag{2.25}$$

which together with (2.14) and (2.16) also gives (2.24). Hence it follows from (2.17), (2.18) and $||d_k|| \le \Delta_k$ that

$$\lim_{k\to\infty} d_k = 0. \tag{2.26}$$

Therefore, we have

$$\begin{aligned} |r_k - 1| &= \left|\frac{Ared_k - Pred_k}{Pred_k}\right| \\ &= \frac{o(||d_k||) + O(||d_k||^2||B_k||)}{Pred_k} \\ &\le \frac{o(||d_k||) + O(||d_k||^2||B_k||)}{||g_k|| \min\{\Delta_k, ||g_k||/||B_k||\}} \\ &\le \frac{o(||d_k||)}{\Delta_k} \\ &\to 0, \end{aligned} \tag{2.27}$$

thus

$$\lim_{k\to\infty} r_k = 1. \tag{2.28}$$

Hence, there are no performs of line search for k sufficiently large. The inequality (2.28) also implies that there exists a positive constant μ^* such that

$$\mu_k > \mu^* \tag{2.29}$$

holds for all sufficiently large k, which gives a contradiction to (2.24). Therefore we see that assumption (2.20) can not be true. The proof is completed.

3. Local Convergence

In the traditional trust region algorithm, the trust region radius will be larger than a positive constant for all sufficiently large k, so the trial step will take the Quasi-Newton step at the end, which guarantees the superlinear convergence of the algorithm. In this section, we show that our new line search and trust region algorithm with the trust region radius converging to zero preserves the superlinear convergence under some certain conditions.

Theorem 3.1. *Suppose the sequence $\{x_k\}$ generated by Algorithm 2.2 converges to x^*, and suppose d_k is the exact solution of subproblem (1.2) and a direction of sufficient descent. If $\nabla^2 f(x)$ is continuous in a neighbourhood of x^* and $\nabla^2 f(x^*)$ is positive definite, and if the condition*

$$\lim_{k\to\infty} ||(\nabla^2 f(x^*) - B_k)d_k||/||d_k|| = 0 \tag{3.1}$$

is satisfied, then the sequence $\{x_k\}$ converges to x^* Q-superlinearly.

Proof. It follows from (3.1) and the positive definiteness of $\nabla^2 f(x^*)$ that there exists a constant $\bar{\delta} > 0$ such that

$$d_k^T B_k d_k \ge \bar{\delta}||d_k||^2 \tag{3.2}$$

holds for all sufficiently large k. Since d_k is a solution of the subproblem (1.2), there exists $\lambda_k \geq 0$ such that

$$g_k + (B_k + \lambda_k I)d_k = 0. \tag{3.3}$$

Define the sets

$$\mathcal{I}_1 = \{k \mid \text{for which } x_{k+1} \text{ is defined by Step 3 of Algorithm 2.2}\}, \tag{3.4}$$

and

$$\mathcal{I}_2 = \{k \mid \text{for which } x_{k+1} \text{ is defined by Step 4 of Algorithm 2.2}\}. \tag{3.5}$$

Then, it follows from (3.2) and (3.3) that for all $k \in \mathcal{I}_2$,

$$Pred_k = \frac{1}{2}d_k^T B_k d_k + \lambda_k ||d_k||^2 \geq \frac{1}{2}\bar{\delta}||d_k||^2, \tag{3.6}$$

and

$$\begin{aligned} Ared_k &= -g_k^T d_k - \frac{1}{2}d_k^T \nabla^2 f(x^*) d_k + o(||d_k||^2) \\ &= Pred_k + \frac{1}{2}d_k^T(B_k - \nabla^2 f(x^*))d_k + o(||d_k||^2) \\ &= Pred_k + o(||d_k||^2). \end{aligned} \tag{3.7}$$

On the other hand, for all $k \in \mathcal{I}_1$, we have from (3.3) that

$$g_k + (B_k + \lambda_k I)\alpha^{-i_k} d_k^{i_k} = 0. \tag{3.8}$$

Then it follows from $\alpha < 1$, $i_k \geq 1$ and (3.2) that the reduction of $f(x)$ from x_k to $x_k + d_k^{i_k}$ is

$$\begin{aligned} Pred_k &= -g_k^T d_k^{i_k} - \frac{1}{2}d_k^{i_k T} B_k d_k^{i_k} \\ &= (\alpha^{-i_k} - \frac{1}{2})d_k^{i_k T} B_k d_k^{i_k} + \alpha^{-i_k}\lambda_k d_k^{i_k T} d_k^{i_k} \\ &> \frac{1}{2}\bar{\delta}||d_k^{i_k}||^2, \end{aligned} \tag{3.9}$$

and the actual reduction from x_k to $x_k + d_k^{i_k}$ is

$$\begin{aligned} Ared_k &= -g_k^T d_k^{i_k} - \frac{1}{2}d_k^{i_k T}\nabla^2 f(x^*) d_k^{i_k} + o(||d_k^{i_k}||^2) \\ &= Pred_k + \frac{1}{2}d_k^{i_k T}(B_k - \nabla^2 f(x^*))d_k^{i_k} + o(||d_k^{i_k}||^2) \\ &= Pred_k + o(||d_k^{i_k}||^2). \end{aligned} \tag{3.10}$$

Combining (3.6)–(3.10), we have

$$\lim_{k\to\infty} r_k = \frac{Ared_k}{Pred_k} = 1. \tag{3.11}$$

Now, we prove that $\{\mu_k\}$ is bounded. Otherwise, $\mu_k \to +\infty$, so the inequality

$$||d_k|| > c_8\Delta_k = c_8\mu_k||g_k|| \tag{3.12}$$

holds for infinitely $k \in \mathcal{I}_2$. The positive definiteness of $\nabla^2 f(x^*)$ indicates that there exist $\hat{M} > \hat{\delta} > 0$ such that

$$\hat{M}||x_k - x^*|| \geq ||g_k|| \geq \hat{\delta}||x_k - x^*||, \tag{3.13}$$

$$\hat{M}||x_k - x^*||^2 \geq f(x_k) - f(x^*) \geq \hat{\delta}||x_k - x^*||^2 \tag{3.14}$$

for all sufficiently large k. Thus, (3.12)–(3.14) show that

$$\begin{aligned} \hat{M}||x_k - x^*||^2 &\geq f(x_k) - f(x^*) \\ &\geq f(x_k + d_k) - f(x^*) \\ &\geq \hat{\delta}||x_k + d_k - x^*||^2 \\ &\geq \hat{\delta}(||d_k|| - ||x_k - x^*||)^2 \\ &\geq \hat{\delta}(c_8\mu_k\hat{\delta} - 1)^2||x_k - x^*||^2, \end{aligned} \tag{3.15}$$

which is impossible if $\mu_k \to +\infty$. Therefore we see that $\{\mu_k\}$ is bounded. This implies

$$||d_k|| \leq c_8\Delta_k \tag{3.16}$$

for all large k. Thus, the trust region is inactive for all large k. Consequently, the superlinear convergence result follows from the standard results of Dennis and Moré.

4. Numerical Results

In this section, we implement our new line search and trust region algorithm with the trust region radius converging to zero (L–NTR), and compare it with three other algorithms: the traditional trust region algorithm (TTR), TTR with line search (L–TTR), and the trust region algorithm with the trust region radius converging to zero without line search (NTR).

The test problems are those given by Moré, Garbow and Hillstrom [3], and we use the same numbering system as that in [3]. In all the tests, the trial step is computed approximately by Algorithm 2.1. The initial approximate Hessian matrix B_1 is chosen as the identy matrix, and B_k is updated by the BFGS formula. However, we do not update B_k if

$$s_k^T y_k > 0 \tag{4.1}$$

fails, where

$$\begin{cases} s_k = x_{k+1} - x_k, \\ y_k = g_{k+1} - g_k. \end{cases} \tag{4.2}$$

In all the tests, we choose $\Delta_1 = \mu_1||g_1||$ with $\mu_1 = 10$. In the tests of TTR and L–TTR, we compute

$$\Delta_{k+1} = \begin{cases} \min\{\dfrac{\Delta_k}{4}, \dfrac{||d_k||}{2}\}, & \text{if } r_k < 0.25, \\ \Delta_k, & \text{if } r_k \in [0.25, 0.75], \\ \max\{4||d_k||, 2\Delta_k\}, & \text{otherwise}. \end{cases} \tag{4.3}$$

In the tests of NTR and L–NTR, we choose $c_2 = c_5 = c_7 = 0.25, c_6 = 10$ and $c_8 = 0.5$. When $f(x_k + d_k) \geq f(x_k)$, we resolve the subproblem (1.2) in the tests TTR and NTR, while we perform the line search in L–TTR and L–NTR. Moreover, we test L–TTR and L–NTR in two versions. In Version 1, we choose $\alpha = 0.1$, while in Version 2, we compute

$$\alpha_k = \max\left\{0.1, \frac{0.5}{1 + (f_k - f(x_k + d_k))/d_k^T g_k}\right\}, \tag{4.4}$$

which is based on truncated quadratic interpolation, set $d_k = \alpha_k d_k$, and repeat this process until a lower function value is obtained.

The algorithm is terminated when the norm of the gradient at the k-th iterate $||g_k||$ is less than $\varepsilon = 10^{-8}$, or when the number of the iterations exceeds $100(n+1)$. The results are listed in the following table. "NF" and "NG" represent the numbers of function calculations and gradient calculations, respectively; if the method fails to find the stational point in $100(n+1)$ iterations, we denote it by the sign "–".

From the table above, we can see that line search is desirable not only for the traditional trust region algorithm but also for the trust region algorithm with trust region converging to zero. Usually we require little computation on the values of the objective function and its gradient when combining trust region and line search techniques. Moreover, algorithm L–NTR often performs better than algorithm L–TTR whatever the version is. Hence, our new line search and trust region algorithm with the trust region converging to zero is more efficient for small size problems.

Results on some of the problems of Moré, Garbow and Hillstrom

		TTR	L–TTR		NTR	L–NTR	
			Version 1	Version 2		Version 1	Version 2
Problem	n	NF/NG	NF/NG	NF/NG	NF/NG	NF/NG	NF/NG
1	3	46/30	37/31	38/30	44/28	44/33	42/31
2	6	42/39	77/72	74/71	48/41	43/40	46/41
3	3	8/6	6/5	7/6	9/6	7/6	7/6
4	2	206/144	301/242	230/188	261/148	263/207	229/181
5	3	34/31	40/37	40/37	43/34	37/32	33/28
6	3	18/12	14/11	14/11	17/10	14/11	14/11
7	9	76/67	75/70	69/64	92/71	79/71	77/68
8	8	249/179	112/93	86/75	251/167	64/54	74/60
9	2	13/11	19/16	15/13	14/10	12/11	12/11
10	2	55/31	31/25	25/20	–	26/18	44/30
12	3	41/36	35/32	42/35	61/39	38/35	46/37
13	6	25/24	28/26	25/24	29/25	30/27	27/26
14	6	90/69	89/79	83/69	127/71	93/72	86/69
15	8	88/78	99/91	84/77	93/74	100/86	79/68
16	2	18/15	17/15	19/17	22/17	20/17	18/15
17	4	55/42	65/55	59/48	142/88	126/97	116/89
18	9	45/33	48/39	38/30	55/31	37/27	40/29

References

[1] J.Y. Fan and Y.X. Yuan, A new trust region algorithm with trust region radius converging to zero, in Proceedings of the 5th International Conference on Optmization: Techniques and Applications, Hong Kong, 2001, 786–794.

[2] D.M. Gay, Computing optimal locally constrained steps, *SIAM J. Sci. Stat. Comp.,* **2** (1981), 186–197.

[3] J.J. Moré, B.S. Garbow and K.H. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software,* **7** (1981), 17–41.

[4] J.J. Moré and D.C. Sorensen, Computing a trust region step, *SIAM J. Sci. Stat. Comp.,* **4** (1983), 553–572.

[5] J. Nocedal and Y.X. Yuan, Combining trust region and line search techniques, in: Y. Yuan, ed., Advances in Nonlinear Programming, Kluwer, 1998, 153–175.

[6] M.J.D. Powell, A new algorithm for unconstrained optimization, in J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., Nonlinear Programming, Academic Press, New York, 1970, 31–66.

[7] T. Steihaug, The conjugate gradient method and trust regions inlarge scale optimization, *SIAM J. Numer. Anal.,* **20** (1983), 626–637.

[8] Y.X. Yuan, Nonlinear Programming: trust region algorithms, in: S.T. Xiao and F. Wu, eds., Proceedings of Chinese SIAM annual meeting, Tsinghua University, Beijing, 1994, 83–97.

[9] Y.X. Yuan, A review of trust region algorithms for optimization, In: J.M. Ball and J.C.R. Hunt, eds., ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, Oxford University Press, 2000, 271–282.

[10] Y.X. Yuan and W.Y. Sun, Optimization Theories and Methods (in Chinese), Science Press, Beijing, 1997.