广西科学 Guangxi Sciences 2006, 13(2):96～101,108

A Nonmonotonic Trust Region Algorithm with Line Search for Unconstrained Optimization *

无约束优化中带线搜索的非单调信赖域算法

MO Jiang-tao, YAN Shi-cui, LIU Chun-yan

莫降涛,颜世翠,刘春燕

(College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, China)

(广西大学数学与信息科学学院,广西南宁 530004)

Abstract: Combining trust region and line search with nonmonotone technique, we give a nonmonotone trust region method for unconstrained optimization. Under suitable conditions, the global convergence and Q-quadratic convergence of our algorithm are well proved. When the trial step is not accepted, we get the next iterative point by nonmonotone line search technique. Unlike traditional nonmonotone algorithms, our method can avoid the possibility that the reference function value used to generate non-monotonicity may be much larger than the real function value. Primary numerical results show that this algorithm is efficient.

Key words: unconstrained optimization, nonmonotonic trust region, line search, global convergence, Q-quadratic convergence

摘要:将信赖域与线搜索方法相结合,采用非单调技术,提出一种求解无约束优问题的非单调信赖域算法,并在适当的条件下,证明算法有全局收敛性和Q-二次收敛性.算法在试探步不被接受时,采用非单调线搜索寻找下一迭代点.算法克服了传统非单调算法中用于产生非单调性的参考函数值远大于实际函数值的问题.初步的数值试验证实算法是有效的.

关键词:无约束优化　非单调信赖域　线搜索　全局收敛　Q-二次收敛

中图法分类号:O211.2　**文献标识码**:A　**文章编号**:1005-9164(2006)02-0096-06

1 Introduction

In this paper, we consider the following unconstrained optimization problem

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

where $f: R^n \to R$ is a continuously differentiable function.

There are two basic approaches to solve the problem (1.1), namely, line search method and trust region method. In the line search method, a computation is performed in a descent direction and a stepsize is found along the direction, see references[1～3]. In the trust region method, there is a region around the current point and a step is chosen to stay in this region where a quadratic model is trusted to be correct. Actually, a trust region method is used to generate a step by the following quadratic model:

$$\min m_k(s) = f_k + g_k^T s + \frac{1}{2} s^T B_k s,$$

$$s.t. \ \|s\| \leqslant \Delta_k, \qquad (1.2)$$

where $f_k \in R$ and $g_k \in R^n$ are the function value and the gradient value of f evaluated at x_k respectively, $B_k \in R^{n\times n}$ is the Hessian matrix of f evaluated at x_k or an approximation to it, Δ_k is the trust region radius and is modified during iterating, based on how well the model agrees with the actual function values. Because of the

收稿日期:2005-09-23

修回日期:2005-12-06

作者简介:莫降涛(1963-),男,广西桂林人,副教授,主要从事最优化理论与方法研究。

* Supported by Science Foundation of Guangxi (No. 0542043).

strong convergence and robustness, many previous studies on the trust region method were reported in references [4～7].

The traditional trust region method for optimization is a monotone method which requires function values decreasing during iterating. However, Grippo et al[3]. found that it was available to find the solution of optimization problems by relaxing the requirement, especially in the presence of steep-sided valleys. Deng et al[5]. proposed a nonmonotonic trust region method which used the nonmonotonic technique proposed by Grippo et al[3]. Numerical results are promising.

But there are two shortcomings for the nonmonotonic algorithms proposed by Deng et al[5]. and Grippo et al[3]. One is that the numerical performances of the algorithms heavily depend on a preset parameter which is used to obtain non-monotonicity. The another one is that the reference function value may be muchlarger than f_k so as to reduce the efficiency of the algorithms. Recently, Dai and Zhang[1] have proposed a new nonmonotonic line search method which can overcome the above disadvantages. Their gradient method with the nonmonotonic line search is effective.

In this paper, we present a nonmonotonic trust region method with line search. It is similar to the method proposed by Nocedal et al[8]. but has two significant differences from that of Nocedal et al[8]. Firstly, in order to obtain the non-monotonicity, the reference function value is amended by the following formulae which are similar to that of Dai et al[1], i. e. ,

if $l_k = \mu$,

$$f_k^r = \begin{cases} f_k^c, \text{if } \dfrac{f_k^{\max} - f_k^{\min}}{f_k^c - f_k^{\min}} > \gamma_1; \\ f_k^{\max}, \text{otherwise}, \end{cases} \quad (1.3)$$

if $p > \upsilon$,

$$f_k^r = \begin{cases} f_k^{\max}, \text{if } f_{k-1}^r > f_k^{\max} > f_k; \\ f_{k-1}^r, \text{otherwise}, \end{cases} \quad (1.4)$$

if $l_k \neq \mu$ and $p \leqslant \upsilon$,

$$f_k^r = f_{k-1}^r, \quad (1.5)$$

where μ and υ are positive integers, l_k is the number of iterations since the current minimal function value is found. Let ω be a prefixed integer, $f_{\max} = \max\{f_{k-j} : 0 \leqslant j \leqslant m(k)\}$, $m(k) = \min\{m(k-1)+1, \omega\}$, $f_k^{\min} = \min\{f_j : 0 \leqslant j \leqslant k\}$. Let x_{k_1} be the point whose function value is equal to $f_k^{\min}$, f_k^c is the maximal function value from x_{k_1} to the point x_k, p is the number of iterations from the latest amending of the reference function value to the iterative point x_k, γ_1 is a prefixed constant. Secondly, when the trial step is not accepted, the performance of a monotone line search but a nonmonotonic one to find the next iterative point will be stopped. In this way, the present nonmonotonic method in performance is better than that in the reference [8]. Primary numerical experiments are encouraging.

This paper is organized as follows. Next section is to describe the algorithm which combines trust region method and nonmonotonic technique with line search. Sections 3 and 4 are the establishment of global convergence and Q-quadratic convergence of the algorithm under suitable conditions. Primary numerical results are presented in Section 5.

2 Algorithm

At iterating k, we generate a trial step s_k to solve the problem (1. 2). In order to reduce the cost of computation, we solve the problem (1. 2) inaccurately such that s_k satisfies

$$\| B_k s_k + g_k \| \leqslant \zeta \| g_k \| \quad (2.1)$$

and

$$m_k(0) - m_k(s_k) \geqslant \delta \| g_k \| \min\{\Delta_k, \| g_k \| / \| B_k \|\}, \quad (2.2)$$

where $\zeta \in (0,1)$, $\delta \in (0,1)$ are constants.

Next we compute r_k, the ratio between the actual reduction $f_k^r - f(x_k + s_k)$ and the predicted reduction $m_k(0) - m_k(s_k)$, i. e. ,

$$r_k = \frac{f_k^r - f(x_k + s_k)}{m_k(0) - m_k(s_k)}, \quad (2.3)$$

where f_k^r is defined by Formulae (1. 3)～(1. 5). If $r_k \geqslant c_0$, we define $x_{k+1} = x_k + s_k$; otherwise, we compute a descent direction d_k satisfying

$$g_k^T d_k \leqslant - a_1 \| g_k \|^2 \quad (2.4)$$

and

$$\| d_k \| \leqslant a_2 \| g_k \|, \quad (2.5)$$

where a_1 and a_2 are two prefixed positive constants. Then along the direction d_k, we search a stepsize α_k satisfying

$$f(x_k + \alpha_k d_k) \leqslant f_k^r + \theta \alpha_k g_k^T d_k \quad (2.6)$$

and

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geqslant \sigma g_k^T d_k, \quad (2.7)$$

where $1/2 < \theta < \sigma < 1$, let $x_{k+1} = x_k + \alpha_k d_k$.

Now, we describe the complete algorithm.

Algorithm 2.1 (The nonmonotonic trust region method).

Step 0 Give $x_0 \in R^n$, $B_0 \in R^{n\times n}$ is a symmetric matrix, the initial radius $\Delta_0 > 0$, $1/2 < \theta < \sigma < 1$, $0 < a_1 < a_2$, $l_0 = 0$, $p = 0$, $\mu \in R$, $\upsilon \in R$, $\omega \in R$, $\gamma_1 \geqslant 1$, $0 < \zeta < 1$, $0 < \delta < 1$, $0 < c_0 < 1$, $0 < c_2 < c_3 < 1 < c_1$, $f_0^{\min} = f_0^r = f_0^{\max} = f_0^c = f_0$, $k = 0$.

Step 1 Compute g_k. If $\| g_k \| = 0$, stop; if $k = 0$, go to Step 3; otherwise, go to Step 2.

Step 2 If $l_k = \mu$, update f_k^r by Formula (1.3), let $l_k = 0$, $p = 0$; otherwise $p = p + 1$.

If $p > \upsilon$, update f_k^r using Formula (1.4); otherwise, $f_k^r = f_{k-1}^r$.

Step 3 Compute an approximate solution s_k of the problem (1.2), such that $\| s_k \| \leqslant \Delta_k$, Formulae (2.1) and (2.2) are satisfied.

Step 4 Compute r_k by Formula (2.3). If $r_k \geqslant c_0$, let $x_{k+1} = x_k + s_k$ and $\Delta_{k+1} \in [\Delta_k, c_1\Delta_k]$, then go to Step 6; otherwise, go to Step 5.

Step 5 Compute a descent direction d_k satisfying Formulae (2.4) and (2.5). Get a stepsize α_k by the line search of Formulae (2.6) and (2.7). Then let $x_{k+1} = x_k + \alpha_k d_k$ and $\Delta_{k+1} \in [c_2 \| s_k \|, c_3\Delta_k]$.

Step 6 If $f_{k+1} < f_k^{\min}$, let $f_{k+1}^{\min} = f_{k+1}^c = f_{k+1}$ and $l_{k+1} = 0$; otherwise, $l_{k+1} = l_k + 1$.

If $f_{k+1} > f_k^c$, let $f_{k+1}^c = f_{k+1}$. Compute $f_{k+1}^{\max}$.

Step 7 Update B_k and set $k = k + 1$. Go to Step 1.

Remark 2.1 In Step 3, s_k can be computed in the same way as that in the reference [5], and, in Step 5, d_k satisfying Formlae (2.4) and (2.5) can be generated by the method in the reference [9].

Remark 2.2 The matrix B_k can be updated by any quasi-Newton formula[8].

3 Global convergence

The following is to analyze the behavior of Algorithm 2.1 when it is applied to the problem (1.1) and some assumptions are required.

Assumption 3.1 The level set $\Omega = \{x \in R^n : f(x) \leqslant f_0\}$ is bounded.

Assumption 3.2 There exists a $\beta > 0$, such that $\| B_k \| \leqslant \beta$ for all k.

Assumption 3.3 There exists a $t > 0$ such that $\| g(x) - g(y) \| \leqslant t \| x - y \|$ for all $x, y \in R^n$.

To be simple, we denote the following notations: $I = \{k : r_k \geqslant c_0\}$, $J = \{k : r_k < c_0\}$, $\beta_k = \Delta_k$ if $k \in I$, $\beta_k = (g_k^T d_k)^2 / \| d_k \|^2$ if $k \in J$, and

$$K_1 = \{q(i) : i \in N, l_{q(i)} = \mu, q(i) < q(i+1)\}, \quad (3.1)$$

where i denotes the i-th point satisfying $l_k = \mu$ in the sequence $\{x_k\}$, $q(i)$ denotes its subscript in the sequence $\{x_k\}$, N is the set of natural numbers.

Lemma 3.1 Let $\{x_k\}$ be the sequence generated by Algorithm 2.1, then $x_k \in \Omega$ for all k.

Proof We prove the lemma by induction. $x_0 \in \Omega$ is obviously true. Suppose that $x_j \in \Omega$, for all j satisfying $0 \leqslant j \leqslant k$. Now we will prove that $x_{k+1} \in \Omega$.

From Formulae (1.3) ~ (1.5), it follows that $f_k^r \in \{f_k^c, f_k^{\max}, f_{k-1}^r\}$. For $f_k^c, f_k^{\max}, f_{k-1}^r$ are the function values of points which belong to the set $\{x_j : 0 \leqslant j \leqslant k\}$, we have that $\max\{f_k^c, f_k^{\max}, f_{k-1}^r\} \leqslant f_0$. Furthermore, $f_k^r \leqslant f_0$.

If $k \in I$, from Formula (2.3), we know that $f_{k+1} < f_k^r$; if $k \in J$, it follows from Formula (2.6) that $f_{k+1} < f_k^r$. Thus, the following inequality holds for all k

$$f_{k+1} < f_k^r. \quad (3.2)$$

This, together with $f_k^r \leqslant f_0$, implies that $x_{k+1} \in \Omega$.

Lemma 3.2 Let $\{x_k\}$ be the sequence generated by Algorithm 2.1, then for all k,

$$f_k^r \geqslant f_k. \quad (3.3)$$

Proof It follows from Formulae (1.3) ~ (1.5) that $f_k^r \in \{f_k^c, f_k^{\max}, f_{k-1}^r\}$. So from Formula (3.2), the definition of f_k^c and $f_k^{\max}$, we have that Formula (3.3) holds for all k.

Lemma 3.3 Suppose that Assumption 3.3 holds and α_k satisfies Formulae (2.4) and (2.5). Then

$$\alpha_k \geqslant ((1-\sigma)/t) |g_k^T d_k| / \| d_k \|^2, \quad (3.4)$$

where σ is defined by Formula (2.7) and t by Assumption 3.3.

Proof The proof is the same as that of Lemma 2.1 in the reference [9].

Next, under some conditions, we will prove that

$$\liminf_{k\to\infty} \beta_k = 0, \quad (3.5)$$

whether K_1, which is defined by Formula (3.1), is finite or infinite.

Lemma 3.4 Suppose that Assumptions 3.1～3.3 hold, and there exists a constant $\varepsilon_0 > 0$ such that $\| g_k \| \geqslant \varepsilon_0$ for all k. If K_1 is a finite set, then Formula (3.5) holds.

Proof Since K_1 is a finite set, we get that $l_k < \mu$ and $p > \upsilon$ when k is sufficiently large, and f_k^r can only be updated by Formula (1.4).

Now, we will prove that

$$\lim_{k\to\infty} \sup f_k = \lim_{k\to\infty} f_k^r. \quad (3.6)$$

From Formulae (1.4) and (3.2), we have that for k sufficiently large,

$$f_{k+2} < f_{k+1}^r \leqslant f_k^r. \quad (3.7)$$

Assumption 3.1 and Formula (3.7) imply that $\{f_k^r\}$ is convergent, and $\lim_{k\to\infty} \sup f_k^r \leqslant f^* = \lim_{k\to\infty} f_k^r$.

Suppose that $\lim_{k\to\infty} \sup f_k < f^*$. Then $\eta = f^* - \lim_{k\to\infty} \sup f_k > 0$. So, for large enough k,

$$f_k \leqslant f^* - \eta + \frac{\eta}{3} = f^* - \frac{2\eta}{3} \leqslant f^* - \frac{\eta}{3} \leqslant f_k^r. \quad (3.8)$$

For sufficiently large k, from Formula (3.8) and the definition of $f_k^{\max}$, we have that $f_{k-1}^r > f_k^{\max}$. If $f_k^{\max} > f_k$, it follows from Formula (1.4) that $f_k^r = f_k^{\max}$, which contradicts Formula (3.8). Thus, $f_k^{\max} = f_k$, that is, when k is large enough, $\{f_k\}$ is monotonically increasing, which should result in $l_k = \mu$. This contradicts that K_1 is a finite set, so Formula (3.6) holds.

Thus, there exists an infinite set K_2, such that the subsequence $\{f_k : k \in K_2\}$ satisfies

$$\lim_{k(\in K_2)\to\infty} f_k = \lim_{k\to\infty} f_k^r. \quad (3.9)$$

For $k+1 \in K_2$, we know that either $k \in I$ or $k \in J$.

Next, we prove the lemma in two cases.

Case 1 $k \in I$, for large enough $k+1 \in K_2$. It follows from Step 4 of Algorithms 2.1 and Formula (2.2), $\| g_k \| \geqslant \varepsilon_0$ and Assumption 3.2 that

$$f_{k+1} \leqslant f_k^r - c_0 \delta \varepsilon_0 \min\{\Delta_k, \varepsilon_0/\beta\}. \quad (3.10)$$

From Formulae (3.9) and (3.10), we obtain that

$$\Delta_k \to 0, \quad (3.11)$$

where $k \in I$ and $k+1 \in K_2$.

Case 2 $k \in J$, for large enough $k+1 \in K_2$. By Formulae (2.6) and (3.4), we know that

$$f_{k+1} \leqslant f_k^r - (\theta(1-\sigma)/t)(g_k^T d_k)^2 / \| d_k \|^2. \quad (3.12)$$

The above inequality, together with Formula (3.9), implies that

$$(g_k^T d_k)^2 / \| d_k \|^2 \to 0, \quad (3.13)$$

where $k \in J$ and $k+1 \in K_2$.

From Formulae (3.11) and (3.13) and the definition of β_k, we conclude that Formula (3.5) holds.

Lemma 3.5 Suppose that Assumptions 3.1～3.3 hold, and there exists a constant $\varepsilon_0 > 0$ such that $\| g_k \| \geqslant \varepsilon_0$ for all k. If K_1 is an infinite set, then Formula (3.5) holds.

Proof For arbitrary $q(i) \in K_1, q(i+1) \in K_1$, when k satisfies $q(i) < k < q(i+1)$, f_k^r can only be updated by Formulae (1.4) and (1.5), thus for $q(i) < k < q(i+1)$,

$$f_k^r \leqslant f_{k-1}^r. \quad (3.14)$$

Now, we prove that there exists an infinite set K_3, such that the subsequence $\{f_k : k \in K_3\}$ satisfies

$$\lim_{k(\in K_3)\to\infty} f_k = \lim_{q(i)\to\infty} f_{q(i)}^{\max}. \quad (3.15)$$

From Formulae (1.3) and (3.14), for all k satisfying $q(i) < k < q(i+1)$, we have that

$$f_k^r \leqslant f_{q(i)}^r \leqslant f_{q(i)}^{\max}. \quad (3.16)$$

It follows from the definition of $f_k^{\max}$, Formulae (3.2), (3.3) and (3.16) that

$$f_{q(i+1)}^{\max} \leqslant \max_{q(i)<k\leqslant q(i+1)} \{f_{q(i)}^{\max}, f_k\} \leqslant \max_{q(i)<k<q(i+1)} \{f_{q(i)}^{\max}, f_k^r\} \leqslant f_{q(i)}^{\max}. \quad (3.17)$$

From Formula (3.17) and Assumption 3.1, $f_{q(i)}^{\max}$ is the function value of some points which belong to the level set Ω, we obtain that $\{f_{q(i)}^{\max}\}$ is convergent. So, there exists an infinite set K_3 such that Formula (3.15) holds.

In addition, for any k, there exists a natural number i, such that $q(i) \leqslant k < q(i+1)$.

Next, we prove the lemma in two cases.

Case 3 $k \in I$, for large enough $k+1 \in K_3$. It follows from Formulae (3.10) and (3.16) that

$$f_{k+1} \leqslant f_{q(i)}^{\max} - c_0 \delta \varepsilon_0 \min\{\Delta_k, \varepsilon_0/\beta\}. \quad (3.18)$$

From Formulae (3.15) and (3.18), the following formula holds

$$\Delta_k \to 0, \quad (3.19)$$

where $k \in I$ and $k+1 \in K_3$.

Case 4 $k \in J$, for large enough $k+1 \in K_3$. From Formulae (3.12) and (3.16), we have that

$$f_{k+1} \leqslant f_{q(i)}^{\max} - (\theta(1-\sigma)/t)(g_k^T d_k)^2 / \| d_k \|^2. \quad (3.20)$$

The above inequality, together with Formula(3.15), implies that

$$(g_k^T d_k)^2 / \| d_k \|^2 \to 0, \quad (3.21)$$

where $k \in J$ and $k+1 \in K_3$.

From Formulae (3.19) and (3.21) and the definition of β_k, we conclude that Formula (3.5) holds.

Theorem 3.1 Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. If Assumptions 3.1~3.3 hold, then

$$\liminf_{k\to\infty} \| g_k \| = 0. \quad (3.22)$$

Proof We will prove the theorem by contradiction. Suppose that there exists a constant $\varepsilon_0 > 0$, such that for all k,

$$\| g_k \| \geqslant \varepsilon_0. \quad (3.23)$$

We denote that $\mu_k = B_k s_k + g_k$. It follows from Formulae(2.1), (3.23) and Assumption 3.2 that

$$\| s_k \| \geqslant \| \mu_k - g_k \| / \| B_k \| \geqslant (\| g_k \| - \| \mu_k \|) / \| B_k \| \geqslant \varepsilon_0 (1-\zeta)/\beta \quad (3.24)$$

holds for all $k \in I$. By Formulae (2.4), (2.5) and (3.23), we know that for all $k \in J$

$$| g_k^T d_k |^2 / \| d_k \|^2 \geqslant a_1^2 \| g_k \|^2 / a_2^2 \geqslant a_1^2 \varepsilon_0^2 / a_2^2. \quad (3.25)$$

The definition of β_k, $\| s_k \| \leqslant \Delta_k$, Formulae(3.24) and (3.25) imply that

$$\beta_k \geqslant \min\{\varepsilon_0 (1-\zeta)/\beta, a_1^2 \varepsilon_0^2 / a_2^2\}. \quad (3.26)$$

It follows from Lemmas 3.4 and 3.5 that Formula (3.26) contradicts Formula(3.5).

4 Local convergence

In this section, we will establish the Q-quadratic convergence of Algorithm 2.1. We suppose that $\{x_k\}$ is a sequence generated by Algorithm 2.1 and the following assumptions hold.

Assumption 4.1 $f(x)$ is twice continuous and $\nabla^2 f(x)$ is Lipschitz continuous in Ω.

Assumption 4.2 x^* is a stationary point of $f(x^*)$ and $\nabla^2 f(x)$ is positive definite.

Assumption 4.3 If B_k is positive definite and $\| B_k^{-1} g_k \| \leqslant \Delta_k$, then $s_k = - B_k^{-1} g_k$.

Obviously, $s_k = - B_k^{-1} g_k$ satisfies Formulae(2.1) and (2.2).

Lemma 4.1 Suppose that Assumptions 3.1, 4.1 and 4.2 hold, and $B_k = \nabla^2 f(x_k)$ for all k. If $\{x_k\}$ converges to x^*, then

$$\lim_{k\to\infty} \| g_k \| = 0. \quad (4.1)$$

Proof By Assumptions 3.1 and 4.1, we get that Assumption 3.2 holds, and Assumptions 3.1 and 4.1 imply that Assumption 3.3 holds. It follows from Theorem 3.1, Assumption 4.1, and $\{x_k\}$ converges to x^*, that the lemma is true.

Lemma 4.2 Suppose that Assumptions 3.1 and 4.1~4.3 hold, and $B_k = \nabla^2 f(x_k)$ for all k. If $\{x_k\}$ converges to x^*, then

$$r_k \geqslant c_0 \quad (4.2)$$

holds for k large enough.

Proof Suppose that k is large enough. Then from Assumption 4.1, B_k is positive definite. If $\| B_k^{-1} g_k \| > \Delta_k$, we have that $\| g_k \| > \Delta_k / \| B_k^{-1} \|$. If $\| B_k^{-1} g_k \| \leqslant \Delta_k$, it follows from Assumption 4.3 that $s_k = - B_k^{-1} g_k$ and $\| g_k \| > \| s_k \| / \| B_k^{-1} \|$. Note that $\Delta_k \geqslant \| s_k \|$, we obtain that

$$\| g_k \| \geqslant \| s_k \| / \| B_k^{-1} \| \quad (4.3)$$

holds for all k. By Formulae (4.1) and (4.3), we know that

$$\lim_{k\to\infty} \| s_k \| = 0. \quad (4.4)$$

Note that $\| B_k^{-1} \| \| B_k \| \geqslant 1$. From Formulae(2.2) and (4.3), it follows that

$$m_k(0) - m_k(s_k) \geqslant \delta \frac{\| s_k \|}{\| B_k^{-1} \|} \min\{\Delta_k, \frac{\| s_k \|}{\| B_k^{-1} \| \| B_k \|}\} \geqslant \delta \frac{\| s_k \|^2}{\| B_k^{-1} \|^2 \| B_k \|}. \quad (4.5)$$

Using Formulae(2.3) and (3.3), we have that

$$r_k \geqslant \frac{f_k - f(x_k + s_k)}{m_k(0) - m_k(s_k)}. \quad (4.6)$$

By Assumptions 4.1 and 4.2, Taylor expansion, $B_k = \nabla^2 f(x_k)$, Formulae(4.4) and (4.5), we deduce that

$$\left| \frac{f_k - f(x_k + s_k)}{m_k(0) - m_k(s_k)} - 1 \right| = \left| \frac{1/2 (s_k^T \nabla^2 f(x_k + \lambda s_k) s_k - s_k^T B_k s_k)}{m_k(0) - m_k(s_k)} \right| \leqslant \frac{1}{2\delta} \| B_k^{-1} \|^2 \| B_k \| \| \nabla^2 f(x_k + \lambda s_k) - B_k \|, \quad (4.7)$$

where $\lambda \in (0,1)$. Thus,

$$\lim_{k\to\infty} \frac{f_k - f(x_k + s_k)}{m_k(0) - m_k(s_k)} = 1. \quad (4.8)$$

From Formulae(4.6) and (4.8), we get that Formula (4.2) holds for sufficiently large k.

The above lemma implies that our algorithm reduces to a nonmonotonic trust region method without line search for k sufficiently large.

Theorem 4.1 Suppose that Assumptions 3.1 and 4.1～4.3 hold, and $B_k = \nabla^2 f(x_k)$ for all k. If $\{x_k\}$ converges to x^*, then Algorithm 2.1 has the Q-quadratic convergence.

Proof We denote that $K = \{k: \| B_k^{-1} g_k \| > \Delta_k\}$. Then $\| g_k \| > \Delta_k / \| B_k^{-1} \|$ for all $k \in K$. This, together with Formula(2.2), indicate that

$$\| g_k \| \| s_k \| + 1/2 \| s_k \|^2 \| B_k \| \geqslant m_k(0) - m_k(s_k) \geqslant \delta \Delta_k^2 / (\| B_k^{-1} \|^2 \| B_k \|). \quad (4.9)$$

Suppose that K is an infinite set. It follows from Formulae(4.1),(4.4) and (4.9) that

$$\lim_{k(\in K) \to \infty} \Delta_k = 0. \quad (4.10)$$

From Lemma 4.2, we know that for sufficiently large k, $\Delta_{k+1} \geqslant \Delta_k$. This contradicts Formula(4.10).

Thus, K is a finite set. From Assumption 4.3, we obtain that $s_k = - B_k^{-1} g_k$ for large enough k, i.e., the step reduces to the Newton step when k is sufficiently large. This means that Algorithm 2.1 reduces to Newton method for large enough k. By Theorem 3.2.1 in the reference [10], Algorithm 2.1 has the property of Q- quadratic convergence.

5 Numerical results

Algorithm 2.1 is implemented and compared with the trust region algorithm combining line search (TRACLS) given by Nocedal et al.[8].

The algorithm is coded in Matlab 6.1[11]. Our program, which is used to find s_k satisfying Formulae (2.1) and (2.2), is similar to Algorithm 4.1 in the reference [5]. The initial trust region radius is chosen as $\Delta_0 = 0.8$. B_k is updated by BFGS formula. When reamending the reference function value, we take $\mu = 4, \upsilon = 20, \omega = 10, \gamma = 10$. If we use line search technique to get the next iterative point, we take $d_k = - g_k$, however, we take $d_k = - B_k^{-1} g_k$, when B_k is positive definite. In all tests, the initial matrix B_0 was chosen as $|f_0| I$, where I is the identify matrix.

The stopping condition is $\max\{\| g_k \|, \| x_k - x_{k-1} \|\} \leqslant 10^{-6}$.

We have tested the algorithms for the problems given by Moré et al[12]. The corresponding numerical results are reported in Table 1, where I, F, G denote the number of iterations, function evaluations and gradient evaluations, respectively; S denotes the number of line searches.

Table 1 Numerical comparisons

Problem	n/m	TRACLS $I/F/G(S//)$	Algorithm 2.1 $I/F/G(S)$
Broyden Tridiagonal	8/8	20/29/21(4)	19/24/23(1)
	16/16	24/45/25(10)	24/29/28(1)
	24/24	32/41/33(4)	29/34/33(1)
	28/28	34/41/35(3)	30/35/34(1)
	32/32	34/41/35(3)	31/36/35(1)
Linear Function-rank 1	12/13	6/53/7(1)	5/6/5(1)
	16/17	6/53/7(1)	5/6/5(1)
	48/49	10/11/11(0)	6/7/6(1)
	52/53	10/11/11(0)	6/7/6(1)
	68/69	11/25/12(3)	6/7/6(1)
	80/81	12/59/13(1)	8/9/8(1)
Linear Function-rank 1 with zero columns and rows	12/13	5/6/6(0)	4/5/4(1)
	56/57	10/11/11(0)	7/8/7(1)
	60/61	11/58/12(1)	6/7/6(1)
	68/69	11/58/12(1)	11/13/10(3)
	72/73	11/58/12(1)	21/25/18(7)
	80/81	12/59/13(1)	24/32/25(7)
Discrete Integral Equation	12/12	21/55/22(9)	19/22/21(1)
	36/36	15/36/16(7)	16/19/18(1)
	52/52	18/37/19(6)	14/17/16(1)
	64/64	17/34/18(6)	14/17/16(1)
	128/128	15/20/16(2)	11/14/13(1)
	256/256	11/12/12(0)	8/11/10(1)
Extended Helical Valley	36/36	44/50/45(2)	44/46/45(1)
	150/150	45/50/46(2)	46/48/47(1)

From Table 1, we find that, for e quite a number of tested problems, Algorithm 2.1 outperforms TRACLS. Therefore, Algorithm 2.1 is an efficient method for unconstrained optimization.

References:

[1] DAI Y H, ZHANG H C. Adaptive two-point stepsize gradient algorithm[J]. Numerical Algorithms, 2001(27): 377-385.

[2] GRIPPO L, LAMPARILLO F, LUCIDI S. A class of nonmonotone stabilization methods in unconstrained optimization[J]. Numerical Mathematics, 1991(59): 779-805.

[3] GRIPPO L, LAMPARILLO F, LUCIDI S. A nonmonotone line search technique for Newton's method[J]. SIAM Journal of Numerical Analysis, 1986, 23(4): 707-716.

[4] CONN A R. NICHOLAS I M, GOULD P L. Trust-region methods [M]. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2000.

(下转第 108 页 Continue on page 108)

4 结束语

模糊多属性决策是一个有前途的研究方向，它在决策科学中的研究相当活跃. 本文针对属性权重信息完全未知或只有部分权重信息且属性值为三角模糊数的模糊多属性决策问题，提出了 3 种基于模糊理想点的最优化决策模型. 通过对这 3 种模型的求解，可获得属性的权重，给出相应的对方案进行排序和择优的决策方法，从而为解决权重信息不完全的模糊多属性决策问题提供了新途径. 实例分析的结果表明，该方法具有可行性和有效性.

参考文献：

[1] DONG W M, WONG F S. Fuzzy weighted averages and implementation of the extension principle[J]. Fuzzy Sets and Systems, 1987, 21: 183-199.

[2] LIOU T S, WANG M J. Fuzzy weighted average: an improved algorithm[J]. Fuzzy Sets and Systems, 1992, 49: 307-315.

[3] LIOU T S, WANG M J. Ranking fuzzy numbers with integral value[J]. Fuzzy Sets and Systems, 1992, 50: 247-255.

[4] GUH Y Y, HONG C C, WANG K M, et al. Fuzzy weighted average: a max-min paired elimination method [J]. Comput Math App, 1996, 32: 115-123.

[5] LEE D H, PARK D. An efficient algorithm for fuzzy weighted average[J]. Fuzzy Sets and Systems, 1997, 87: 39-45.

[6] KAO C, LIU S T. Fractional programming approach to fuzzy weighted average [J]. Fuzzy Sets and Systems, 2001, 120: 435-444.

[7] 徐泽水. 对方案有偏好的三角模糊数型多属性决策方法研究[J]. 系统工程与电子技术, 2002, 24(8): 9-12.

[8] 徐泽水. 基于期望值的模糊多属性决策法及其应用[J]. 系统工程理论与实践, 2004(1): 109-113.

[9] 汪培庄. 模糊集合理论及其应用[M]. 上海: 科学技术出版社, 1983.

（责任编辑：邓大玉）

（上接第 101 页 Continue from page 101）

[5] DENG N Y, XIAO Y, ZHOU F J. Nonmonotonic trust region algorithm[J]. Journal of Optimization Theory and Applications, 1993, 76(2): 259-285.

[6] FU J H, SUN W Y. Nonmonotone adaptive trust-region method for unconstrained optimization problems [J]. Applied Mathematics and Computation, 2005, 163: 489-504.

[7] SUN W, SAMPAIO R J B, YUAN J Y. Trust region methods with quasi-Newton update for nonsmooth least-square problems [J]. Applied Mathematics and Computation, 1999, 105: 183-194.

[8] NOCEDAL J, YUAN Y. Combining trust region and linear search techniques [M]//Yuan Y. Advances in nonlinear programming. Dordrecht: Kluwer Academic Publishers, 1998: 153-175.

[9] ZHANG H C, HAGER W W. A nonmonotone line search technique and its application to unconstrained optimization[J]. SIAM J Optim, 2004, 14(4): 1043-1056.

[10] 袁亚湘，孙文瑜. 最优化理论和方法[M]. 北京: 科学出版社, 1997.

[11] 飞思科技产品研发中心. MATLAB 6.1 辅助优化计算与设计[M]. 北京: 电子工业出版社, 2003.

[12] MORé J J, GARBOW B S, HILLSTROM K E. Testing unconstrained optimization [J]. ACM Trans Math Software, 1981, 7: 17-41.

（责任编辑：邓大玉）