Pure Mathematics 理论数学, 2013, 3, 312-316
http://dx.doi.org/10.12677/PM.2013.35048 Published Online September 2013 (http://www.hanspub.org/journal/pm.html)

A Nonmonotonic Self-Adaptive Trust Region Algorithm*

Dan Hang, Xiaoyan Wang, Jianzhong Hao, Ya Wang

Department of Basic Education, Air Force Logistics College, Xuzhou
Email: hazel-hang@sohu.com

Received: Jun. 18th, 2013; revised: Jul. 24th, 2013; accepted: Aug. 3rd, 2013

Abstract: We propose an improved nonmonotonic trust region algorithm. Our method is to use the nonmonotone technique and if the trial step is rejected, the stepsize is computed by a fixed formula. The trust region radius is updated at a variable rate. Numerical experiment results show that the new algorithm is effective. Under mild conditions, we prove that the algorithm is global convergence.

Keywords: Unconstrained Optimization; Trust Region; Fixed Stepsize; Nonmonotonic Technique; Global Convergence

一种非单调自适应信赖域法*

杭　丹，王晓燕，郝建忠，王　娅

空军勤务学院基础部，徐州
Email: hazel-hang@sohu.com

收稿日期：2013 年 6 月 18 日；修回日期：2013 年 7 月 24 日；录用日期：2013 年 8 月 3 日

摘　要：求解无约束优化问题，本文给出了一种改进的非单调信赖域算法。采用了非单调技术，当试探步不被接受时，下一步的迭代步长由一个固定公式给出。并直接给出调整信赖域半径公式。数值试验结果表明新算法的有效性。在适当的条件下，证明了算法的全局收敛性。

关键词：无约束最优化；信赖域；固定步长；非单调技术；全局收敛性

1. 引言

求解无约束优化问题：

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

其中 $f(x): R^n \to R^1$ 是二阶连续可微的函数,信赖域方法是解决问题(1)一类有效的迭代方法，而且保证了很强的收敛性。在每个迭代点 x_k，通过解如下的子问题产生一个试探步 d_k：

$$\min \phi_k(d) = g_k^T d + \frac{1}{2} d^T B_K d \tag{2}$$

$$\text{s.t.}\quad \|d\| \le \Delta_k \tag{3}$$

其中 g_k 是目标函数在当前点 x_k 处的梯度。B_k 为精确 Hessian 阵或其近似 $\Delta_k > 0$ 是信赖域的半径。

*基金项目：安徽省教育厅自然科学研究项目(编号：KJ2012B125)，安徽省高等学校省级优秀青年人才基金项目(编号：2012SQRL155)。

d_k 是子问题(2)~(3)的一个精确或近似解，得到 d_k 后计算比率：

$$r_k = \frac{\mathrm{Ared}_k}{\mathrm{Pred}_k} = \frac{f(x_k) - f(x_k + d_k)}{\phi_k(0) - \phi_k(d_k)} \tag{4}$$

检验试探步 d_k 是否被接受，然后信赖域半径 Δ_k 根据 r_k 值调整。对于传统的信赖域方法，信赖域半径 Δ_k 在给定的比率下调整，然而，HEI LONG[1]中提出了自适应信赖域方法，其中 Δ_k 根据 r_k 的一个函数值进行调整，数值结果显示是有效的。本文在其启示下，改进 Hei Long[1]中调整信赖域半径的公式即采用 $\Delta_{k+1} = R_{c2}(r_k)\Delta_k$ 公式进行调整信赖域半径；同时采用了非单调技术，且当试探步 d_k 不被接受时，不再重解子问题，而是通过一个固定的公式 $x_{k+1} = x_k - \dfrac{\delta g_k^T d_k}{d_k{}^T B_k d_k} d_k$ 直接得到下一个迭代点。数值试验表明，这样做大大减少了计算量。在适当的条件下，证明了新算法的全局收敛性。

2. 算法[2]：算法中 d_k 的选取有多种方法(如文[2]-[4]).

定义 1 设一维函数 $R_\eta(t)$ 定义在 $R = (-\infty, +\infty)$，其中 $\eta \in (0,1)$，$R_\eta(t)$ 是 R-函数当且仅当它满足：

1) $R_\eta(t)$ 在 $(-\infty, +\infty)$ 不是下降的；
2) $\lim\limits_{t \to -\infty} R_\eta(t) = \beta_1$，其中 $\beta_1 \in (0,1)$ 是小常数；
3) $R_\eta(t) \le 1 - \gamma_1$，对所有 $t < \eta$，这里 $\gamma_1 \in (0, 1-\beta_1)$ 是一常数；
4) $R_\eta(t) = 1 + \gamma_2$，其中 $\gamma_2 \in (0, +\infty)$ 是一常数；
5) $\lim\limits_{t \to +\infty} R_\eta(t) = \beta_2$，其中 $\beta_2 \in (1+\gamma_2, +\infty)$ 是一常数。

定理 2 R-函数 $R_\eta(t)$ (其中 $\eta \in (0,1)$)满足：$0 < \beta_1 \le R_\eta(t) \le 1 - \gamma_1 < 1, \forall t \in (-\infty, \eta)$，$1 < 1 + \gamma_2 \le R_\eta(t) \le \beta_2 < \infty, \forall t \in [\eta, +\infty)$。求解子问题(2)~(3)获得试探步 d_k 满足下面条件：

$$\phi_k(0) - \phi_k(d_k) \ge \Gamma \|g_k\| \min\left\{\Delta_k, \|g_g\| / \|B_k\|\right\} \tag{5}$$

$$d_k^T g_k \le -\Gamma \|g_k\| \min\left\{\Delta_k, \|g_g\| / \|B_k\|\right\} \tag{6}$$

其中 $\Gamma \in (0,1)$ 是一个常数。采用非单调技术，设 $f_{l(k)} = f(x_{l(k)}) = \max\limits_{0 \le j \le m(k)} f(x_{k-j})$ 这里 $m(0) = 0$，$m(k) = \min\{m(k-1)+1, M\}$，$M$ 是一个非负整数。故获得试探步实际下降量 $\mathrm{Ared}_k = f_{l(k)} - f(x_k + d_k)$，由(4)计算比率，检验试探步 d_k 是否被接受，当试探步 d_k 不被接受时，不再重解子问题，而是用固定公式直接给出步长 $x_{k+1} = x_k + \alpha_k d_k = x_k - \dfrac{\delta g_k^T d_k}{d_k^T B_k d_k} d_k$，然后信赖域半径 Δ_k 根据 r_k 值调整。

算法：

步 0。给定 $x_0, B_0, \Delta_0 > 0, \varepsilon \ge 0, 0 < \beta_1 < 1, 0 < \gamma_1 < 1 - \beta_1, \gamma_2 > 0, \beta_2 > 1 + \gamma_2$，$0 < c_1 < c_2 < 1$，$M > 0$，$\delta \in (0,1), \mu > 0, \omega > 1$。置 $k = 0, m(0) = 0$。

步 1。计算 $g_k = g(x_k)$。如果 $\|g_k\| \le \varepsilon$，则终止。否则计算 B_k 和 $f_{l(k)}$。

步 2。解子问题(2)~(3)使 d_k 满足(4)~(5)。

步 3。计算 Ared_k，Pred_k，和 $r_k = \dfrac{\mathrm{Ared}_k}{\mathrm{Pred}_k}$。

步 4。如果 $r_k > c_1$，转 Step 5。否则计算 $\alpha_k = -\dfrac{\delta g_k^T d_k}{d_k^T B_k d_k}$，置 $x_{k+1} = x_k + \alpha_k d_k$，转 Step 6。

步 5。置 $x_{k+1} = x_k + d_k$。步 6。$\Delta_{k+1} = R_{c2}(r_k)\Delta_k$。置 $m(k+1) = \min\{m(k)+1, M\}, k = k+1$，转 Step 1。

注：我们注意到在[1]中信赖域半径由 $\Delta_{k+1} = R_{c2}(r_k)\|d_k\|_2$ 确定。但当 d_k 很小时会造成下一步迭代的信赖域半

径太小，使下一步试探步长过小，影响算法的效率.因此在我们的算法中采用 $\Delta_{k+1}=R_{c2}\left(r_k\right)\Delta_k$ 公式进行迭代来避免这种缺陷。事实上，后面的数值试验也验证了这种修改的有效性。

3. 收敛性

假设 1。1) 函数 $f\left(x\right)$ 在 R^n 是 LC^1 即存在 $\mu>0$ 有 $\left\|g\left(x\right)-g\left(y\right)\right\|\le\mu\left\|x-y\right\|, \forall x,y\in R^n$，其中

$$g\left(x\right)=\nabla f\left(x\right) \tag{7}$$

2) 给定 x_0 属于 R^n，水平集 $\Omega_0=\left\{x\in R^n\left\|f\left(x\right)\le f\left(x_0\right)\right.\right\}$ 是紧集。

3) 矩阵 B_k 正定且存在实数 ω 使得

$$d^T B_k d\ge\omega d^T d,\ \forall d\in R^n\quad k=0,1,\cdots \tag{8}$$

定义两个指标集：$I=\left\{k:r_k\ge c_1\right\}$，$J=\left\{k:r_k<c_1\right\}$。

引理 2[3]假设 $\left\{x_k\right\}$ 是由算法产生的序列.如果假设 3.1 满足且有

$$\delta\in\left(0,\omega/\mu\right). \tag{9}$$

则

$$f\left(x_{k+1}\right)-f_{l(k)}\le\left\{\left(1-\mu\delta/\omega\right)\delta/2\right\}g_k^T d_k,\ \forall k\in J \tag{10}$$

引理 3[3] 算法产生的序列 x_k 定义在 Ω_0。如果假设 3.1 和(8)成立，则 $f_{l(k)}$ 是下降的。

引理 4 如果假设 3.1 和(9)满足，且存在 $\varepsilon>0$ 使得对所有的 k 有 $\left\|g\left(x\right)\right\|\ge\varepsilon$，那么存在一个常数 $\gamma>0$ 有 $\Delta_k\ge\gamma/M_k$，$k=0,1,\cdots$

这里 M_k 定义为

$$M_k=\max_{0\le i\le k}\left\|B_i\right\|+1\text{。} \tag{11}$$

证明：令 $\sigma=\Gamma\delta\left(1-c_2\right)/2\mu$，由 $\nabla f\left(x\right)$ 是一致连续的，对任意 $x_k\in\Omega_0$，$\theta_k\in\left[0,1\right]$，$\left\|d_k\right\|<\sigma$ 有

$$d_k\left[\nabla f\left(x_k+\theta_k d_k\right)-\nabla f\left(x_k\right)\right]\le\mu\left\|d_k\right\|^2<\mu\Gamma\left(1-c_2\right)\varepsilon\left\|d_k\right\|/2\mu=\Gamma\left(1-c_2\right)\varepsilon\left\|d_k\right\|/2 \tag{12}$$

用归纳法证明(11)成立。设

$$\gamma=\min\left\{\Delta_0 M_0,\beta_1\sigma M_0,\beta_1\varepsilon,\Gamma\left(1-c_2\right)\beta_1\varepsilon\right\} \tag{13}$$

当 $k=0$, 由(13)知 $\Delta_0\ge\gamma/\mathrm{M}_0$ 即(11)对 $k=0$ 成立。假设(11)对 k 成立，因 M_k 递增，只要证明：

$$\Delta_{k+1}\ge\gamma/\mathrm{M}_k \tag{14}$$

当 $r_k\ge c_2$ 时，由 $\Delta_k=R_{c_2}\left(r_k\right)\Delta_k$ 知 $\Delta_{k+1}\ge\Delta_k\ge\gamma/\mathrm{M}_k$，故(14)成立。

当 $r_k<c_2$ 时，知 $\Delta_{k+1}\ge\beta_1/\Delta_k$ 如果 $\left\|d_k\right\|\ge\sigma$，则 $\Delta_{k+1}\ge\beta_1\Delta_k\ge\beta_1\left\|d_k\right\|\ge\beta_1\sigma\ge\beta_1\sigma M_0/M_k\ge\gamma/M_k$。

设 $d_k<\sigma,r_k<c_2$，得

$$f\left(x_k+d_k\right)-f_{l(k)}>-c_2\left(\phi_k\left(0\right)-\phi_k\left(d_k\right)\right)=c_2\left(g_k^T d_k+d_k{}^T B_k d_k/2\right) \tag{15}$$

$$f\left(x_k+d_k\right)-f_{l(k)}\le g\left(\eta\right)^T d_k=g_k^T d_k+\left(g\left(\eta\right)-g_k\right)^T d_k\le g_k^T d_k+\mu\left\|d_k\right\|^2\le g_k^T d_k+\Gamma\left(1-c_2\right)\left\|d_k\right\|/2 \tag{16}$$

其中 $\eta=x_k+\theta_k d_k\in\left[x_k,x_k+d_k\right]$。根据(14)和(15)，得到

$$\left(1-c_2\right)\left(g_k^T d_k+\Gamma\varepsilon\left\|d_k/2\right\|\right)>c_2 d_k^T B_k d_k/2 \tag{17}$$

此外，由(5)和 $\left\|g_k\right\|\ge\varepsilon$ 得到

$$-g_k^Y d_k - d_k^Y B_k d_k/2 \ge \Gamma\varepsilon \min\{\Delta_k, \varepsilon/\|B_k\|\} \tag{18}$$

上式两边乘$1-c_2$与(17)相加得到

$$\begin{aligned}\Delta_k\|B_k\| &\ge -d_k^T B_k d_k > 2(1-c_2)\Gamma\varepsilon\min\{\Delta_k,\varepsilon/\|B_k\|\} - (1-c_2)\Gamma\varepsilon\Delta_k \ge (1-c_2)\Gamma\varepsilon\min\{\Delta_k,\varepsilon/\|B_k\|-\Delta_k\}\\ &= (1-c_2)\Gamma\varepsilon\min\{\Delta_k, 2\varepsilon/\|B_k\|-\Delta_k\}\end{aligned} \tag{19}$$

于是

$$\Delta_k\|B_k\| \ge (1-c_2)\Gamma\varepsilon\min\{1, 2\varepsilon/\|B_k\|-1\} \tag{20}$$

若$\Delta_k\|B_k\| \le \varepsilon$，则$\Delta_k\|B_k\| > (1-c_2)\Gamma\varepsilon$，否则$\Delta_k\|B_k\| > \varepsilon$，因而由(16)有

$$\Delta_k\|B_k\| \ge \min\{(1-c_2)\Gamma\varepsilon, \varepsilon\} \ge \gamma/\beta_1, \tag{21}$$

于是$\Delta_{k+1} > \beta_1\Delta_k > \gamma/\|B_k\| > \gamma/M_k$。

定理 5 如果假设 1 和(9)满足，同时$\{B_k\}$满足

$$\sum_{k=1}^{\infty} 1/M_k = \infty \tag{22}$$

这里$M_k = \max_{0\le i\le k}\|B_i\|+1$，则由算法产生的序列$\{x_k\}\subset\Omega_0$，且满足

$$\lim_{k\to\infty}\inf\|g_k\| = 0 \tag{23}$$

证明：假设(23)不成立，则存在$\varepsilon > 0$使得对任意k均有$\|g_k\| \ge \varepsilon > 0$。当$k\in I$时，由(4)和(6)得到

$$f_{l(k)} - f(x_{k+1}) > c_1(\phi_k(0) - \phi_k(d_k)) = c_1\Gamma\varepsilon\min\{\Delta_k, \varepsilon/\|B_k\|\} \tag{24}$$

当$k\in J$时，由(6)和(10)得到

$$f_{l(k)} - f(x_{k+1}) \ge \{1-\mu\delta/\omega)\delta/2\}(-g_k^T d_k) \ge \{(1-\mu\delta/\omega)\delta/2\}\Gamma\varepsilon\min\{\Delta_k, \varepsilon/\|B_k\|\} \tag{25}$$

令$\theta = \min\{c_1\Gamma\varepsilon, (1-\mu\delta/\omega)\delta/2)\}$，则由(24)和(25)得到对任意$k$都有

$$f_{l(k)} - f(x_{k+1}) \ge \theta\min\{\Delta_k, \varepsilon/\|B_k\|\} \tag{26}$$

由引理 3 知对任意k均有

$$\min\{\Delta_k, \varepsilon/\|B_k\|\} \ge \nu/M_k \tag{27}$$

其中$\nu = \min\{\gamma, \varepsilon\}$. 由$f_{l(k)}$递减及(26)和(27)知，对$s = 0,1,\cdots,M$，均有

$$f_{l(k)} \ge f_{l(k+s)} \ge f_{k+s+1} + \theta\min\{\Delta_k, \varepsilon/\|B_{k+s}\|\} \ge f_{k+s+1} + \theta\nu/M_{k+s}\} \tag{28}$$

根据引理 2 和M_k递增知

$$f_{l(k)} \ge \max\{f_{k+s+1} : 0 \le s \le M\} + \theta\nu/M_{k+M+1} = f_{k+M+1} + \theta\nu/M_{k+M+1}\} \tag{29}$$

再根据假设 1 和引理 2，即$f_{l(k)}$是单调下降且收敛的故由(29)得

$$\begin{aligned}\sum_{k=1}^{\infty} 1/M_{k+M+1} &\le (1/\nu\theta)\sum_{k\ge1}\left(f_{l(k)} - f_{l(k+M+1)}\right)\\ &= (1/\nu\theta)\sum_{k\ge1}\sum_{s=0}^{M}\left(f_{l(k+s)} - f_{l(k+s+1)}\right) \le (M/\nu\theta)\sum_{k\ge1}\left(f_{l(k)} - f_{l(k+1)}\right) < \infty\end{aligned} \tag{30}$$

显然(30)和(22)矛盾，定理得证。

4. 数值结果

本节中，算法用[5]的标准无约束优化问题测试。使用和[5]同样的编号。此外，假设 $B_k = H_k$，系数的选择为：$\varepsilon = 10^{-8}, c_1 = 0.01, c_2 = 0.25. \beta_1 = 0.1, \beta_2 = 5, \delta = 0.1$ 。$\gamma_1 = \gamma_2 = 0.15, M = 10$ 。

其中用 n 表示问题的维数，用 ng 表示梯度迭代次数，nf 表示函数值的迭代维数，f 表示最终的函数值，L 是标准初始点的常用对数因子，即，如果 x_0 是标准初始点， 则实际的初始点为 $10^L \cdot x_0$.算法与黑龙[1]进行比较. 结果在表 1 中，可以发现我们的算法更有效。

Table 1. Numerical results
表 1. 数值结果

NO.	n	L	本文算法(M = 10)			文[2]算法		
			nf	ng	f	nf	ng	f
8	50	0	34	34	4.3179e−004	46	46	4.3179e−004
	100	0	37	37	9.0249e−004	47	47	9.0249e−004
	200	0	41	41	0.0019	53	53	0.0019
14	50	0	16	15	2.1526e−022	27	27	0
	100	0	15	14	7.11102e−023	27	27	0
	200	0	19	17	0	27	27	0

参考文献 (References)

[1] Long Hei, A self-adaptive trust regional gorithm, Journal of Computational Mathematics, 2003, 21: 229-236.
[2] J. Nocedal, Y. Yuan, Combing trust region and line search techniques. In: Y. Yuan, Ed., Advances in Nonlinear Programming, Kluver: Springer, 1998: 153-175.
[3] J. T. Mo, K. Zhang and Z. X. Wei. A nonmonotone trust region method for unconstrained optimization. Applied Mathematics and Computation, 2005, 171(1): 371-384.
[4] L. Grippo, F. Lampariello and S. Lucidi. A nonmonotonic line search technique for Newton’s method. SIAM Journal on Numerical Analy- sis, 1986, 23(4): 707-716.
[5] J. J. Mor, B. S. Garbow and K. E. Hillstrom. Testing unconstrained optimization software. ACM Transactions on Mathematical Software, 1981, 7(1): 17-41.