A New Nonmonotone Gradient-Path Algorithm for Unconstrained Optimization

Zhang Chun[1]，Sun Wenyu[2]，Chen Jun[3]，Zhang Gui[1]

(1. Institute of Science，PLA University of Science and Technology，Nanjing 211101，China)
(2. School of Mathematical Sciences，Nanjing Normal University，Nanjing 210046，China)
(3. School of Mathematics & Information Technology，Nanjing Xiaozhuang University，Nanjing 211171，China)

Abstract: This paper presents a nonmonotone gradient-path algorithm by approximating the secant equation for unconstrained optimization problem. The nonmonotone criterion is used to speed up the convergence progress of objective function. Theoretical analysis is given which proves that the proposed algorithm is weakly globally convergent. The results of numerical experiments are reported to show the effectiveness of the proposed algorithm.

Key words: unconstrained optimization，gradient-path，nonmonotone technique，global convergence
CLC number: O242.2 **Document code:** A **Article ID:** 1001-4616(2011)03-0001-06

一种新的非单调梯度路径线搜索方法

张 纯[1]，孙文瑜[2]，陈 俊[3]，张 瑰[1]

(1. 中国人民解放军理工大学理学院，江苏 南京 211101)
(2. 南京师范大学数学科学学院，江苏 南京 210046)
(3. 南京晓庄学院数学与信息技术学院，江苏 南京 211171)

[摘要] 通过近似处理割线方程提出一种解无约束优化问题的单调梯度路径算法. 其中，非单调技术用于加速目标函数的收敛过程. 理论分析给出了算法的弱全局收敛性，数值结果表明了算法的有效性.

[关键词] 无约束优化，梯度路径，非单调技术，全局收敛性

In this paper we consider a new gradient-path method for large scale unconstrained minimization problem

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

where f is a real-valued function on $\mathbf{R}^n$. For any given x_k, this method uses the quadratic model

$$\psi_k(\omega) = f_k + g_k^{\mathrm{T}} \omega + \frac{1}{2}\omega^{\mathrm{T}} B_k \omega, \tag{1}$$

where we assume throughout that both the gradient $g(x) = \nabla_x f(x)$ and the matrix $B(x) = \nabla_{xx} f(x)$ of f exist. Via a solution of the differential equation

$$\dot{\gamma}(t) = -\nabla\psi(\gamma(t)), \quad \gamma(0) = 0,$$

we can get a gradient-path of the quadratic model $\psi(x)$ in [1]. However，calculation of B_k and its inverse Hessian matrix is usually time-consuming and impractical. So we can modify the gradient-path by approximating the secant equation.

Received date: 2011-06-27.
Foundation item: Supported by the National Natural Science Foundation of China(10871098，11071122)，the Special Research Foundation of Doctoral Program of Higher Education of China(20103207110002)，the Advanced Roserch Foundation of PLA University of Science and Technology (20110516).
Corresponding author: Sun Wenyu，professor，majored in numerical mathematics. E-mail: wysun@njnu.edu.cn

The paper is organized as follows. In Section 1, we define the new gradient path by approximating the secant equation and its property. In Section 2, we describe the nonmonotone gradient-path algorithm. In Section 3, the weak global convergence is established. Finally, the results of numerical experiments of the new proposed gradient-path algorithm is reported in Section 4.

1 A New Modified Gradient Path

The solution of the differential equation

$$\dot{\boldsymbol{\gamma}}(t) = -\nabla\psi(\boldsymbol{\gamma}_{(t)}), \quad \boldsymbol{\gamma}(0) = 0 \tag{2}$$

is named as the gradient-path of the quadratic function $\psi(\boldsymbol{\omega})$. It can be given in the following closed form[1]:

$$\boldsymbol{x}(t) = \boldsymbol{x}_k + (\mathrm{e}^{-tB_k} - I)\boldsymbol{B}_k^{-1}\boldsymbol{g}_k. \tag{3}$$

In [2], an explicit calculation of stepsize α_k is presented, where α_k is obtained by minimizing $\|\alpha\boldsymbol{\Delta x} - \boldsymbol{\Delta g}\|$ with $\boldsymbol{\Delta x} = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\boldsymbol{\Delta g} = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$, and α_k is given as

$$\alpha_k = \langle \boldsymbol{\Delta x}, \boldsymbol{\Delta g}\rangle / \langle \boldsymbol{\Delta x}, \boldsymbol{\Delta x}\rangle, \tag{4}$$

where $\langle \boldsymbol{a}, \boldsymbol{b}\rangle$ denotes the scalar product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

In (3), if we use $\alpha_k I$ to approximate B_k, we yield the gradient-path

$$\boldsymbol{\gamma}(t) = (\mathrm{e}^{-t\alpha_k I} - I)\frac{1}{\alpha_k}I\boldsymbol{g}_k.$$

For the composite function $\mathrm{e}^{-t\alpha_k I}$, we have

$$\boldsymbol{\gamma}(t) = (\mathrm{e}^{-t\alpha_k I} - I)\frac{1}{\alpha_k}I\boldsymbol{g}_k = \left(\sum_{p=0}^{\infty}\frac{(-t\alpha_k I)^p}{p!} - I\right)\frac{1}{\alpha_k}\boldsymbol{g}_k = \left(\sum_{p=0}^{\infty}\frac{(-t\alpha_k)^p}{p!} - I\right)\frac{1}{\alpha_k}\boldsymbol{g}_k = (\mathrm{e}^{-t\alpha_k} - 1)\frac{1}{\alpha_k}\boldsymbol{g}_k. \tag{5}$$

Therefore, we can use $\boldsymbol{\gamma}(t)$ in (5) as a new gradient-path.

We summarize the property of the modified gradient-path as follows.

Lemma 1 Let the step $\boldsymbol{\gamma}_k(t)$ be obtained from the new gradient-path, we assume that $\alpha_k > 0$. Then we have that the norm function of the path is monotonically increasing for $t \in [0, +\infty)$, and that the function $\psi_k(\boldsymbol{\omega})$ is monotonically discreasing for $t \in (0, +\infty)$. Furthermore,

$$\boldsymbol{g}_k^{\mathrm{T}}\frac{\mathrm{d}\boldsymbol{\gamma}_k(t)}{\mathrm{d}t} \to -\|\boldsymbol{g}_k\|^2 \quad \text{as} \quad t \to 0. \tag{6}$$

Proof From the definition of new path (5), we have

$$\|\boldsymbol{\gamma}_k(t)\| = \|(\mathrm{e}^{-t\alpha_k} - 1)\frac{1}{\alpha_k}\boldsymbol{g}_k\|.$$

Then using $\alpha_k > 0$ and $\mathrm{e}^{-t\alpha_k} < 1$, we have

$$\|\boldsymbol{\gamma}_k(t)\| = (1 - \mathrm{e}^{-t\alpha_k})\frac{1}{\alpha_k}\|\boldsymbol{g}_k\|.$$

Let

$$\varphi_k(t) = \|\boldsymbol{\gamma}_k(t)\| = (1 - \mathrm{e}^{-t\alpha_k})\frac{1}{\alpha_k}\|\boldsymbol{g}_k\|,$$

we have

$$\varphi'_k(t) = -(-\alpha_k)\mathrm{e}^{-t\alpha_k}\frac{1}{\alpha_k}\|\boldsymbol{g}_k\| = \mathrm{e}^{-t\alpha_k}\|\boldsymbol{g}_k\| > 0,$$

which means that $\|\boldsymbol{\gamma}_k(t)\|$ is monotonically increasing for $t \in [0, +\infty)$. From

$$\psi_k(t) = f_k + \boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\gamma}_k(t) + \frac{1}{2}\boldsymbol{\gamma}_k(t)^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{\gamma}_k(t),$$

with $\boldsymbol{B}_k = \alpha_k \boldsymbol{I}$, we know that

$$\frac{\mathrm{d}\psi_k(t)}{\mathrm{d}t} = \boldsymbol{g}_k^{\mathrm{T}}\frac{\mathrm{d}\boldsymbol{\gamma}_k(t)}{\mathrm{d}t} + \alpha_k\boldsymbol{\gamma}_k^{\mathrm{T}}\frac{\mathrm{d}\boldsymbol{\gamma}_k(t)}{\mathrm{d}t} = -\mathrm{e}^{-t\alpha_k}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_k - \mathrm{e}^{-t\alpha_k}(\mathrm{e}^{-t\alpha_k} - 1)\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_k = -\mathrm{e}^{-2t\alpha_k}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_k \leqslant 0.$$

which means that $\psi_k(t)$ is monotonically descreasing for $t \in (0, +\infty)$.

Further, from

$$g_k^{\mathrm{T}} \frac{\mathrm{d}\gamma_k(t)}{t} = -\mathrm{e}^{-t\alpha_k} \| g_k \|^2,$$

we have

$$g_k^{\mathrm{T}} \frac{\mathrm{d}\gamma_k(t)}{\mathrm{d}t} \to -\| g_k \|^2 \quad \text{as} \quad t \to 0. \tag{7}$$

2 The New Gradient Path Algorithm

The nonmonotone technique is an efficient one for optimization (see [3-10]). In this section we describe an algorithm with new gradient-path (5) and the nonmonotone technique.

Algorithm 1

Step 0 Initialization Step.

Given x_0 and α_0. Choose parameters $\eta \in \left(0, \frac{1}{2}\right)$, $\omega \in (0,1)$, $\delta > 0$, $0 < \sigma < 1$, $\varepsilon > 0$ and positive integer M. Set $m(0) = 0$ and $k = 0$.

Step 1 Test for convergence.

Compute g_k. If $\| g_k \| \leqslant \varepsilon$, stop.

Step 2 If $\alpha_k \leqslant \sigma$ or $\alpha_k \geqslant \frac{1}{\sigma}$, set $\alpha_k = \delta$.

Step 3 Choice of x_{k+1}.

Choose $t_k = \infty, \omega^{-n}, \omega^{-(n-1)}, \cdots$, and compute $\gamma(t_k) = (\mathrm{e}^{-t_k\alpha_k} - 1)\frac{1}{\alpha_k} g_k$ until the following inequality is satisfied

$$f(x_k + \gamma_k) \leqslant f(x_{l(k)}) + \eta(1 - \mathrm{e}^{-t_k}) < \gamma'(0), g_k >, \tag{8}$$

where $f(x_{l(k)}) = \max_{0 \leqslant j \leqslant m(k)-1} \{ f(x_{k-j}) \}$.

Step 4 Compute α_{k+1}.

Set $\alpha_{k+1} = -(g_k^{\mathrm{T}} y_k)/(\alpha_k g_k^{\mathrm{T}} g_k)$, where $y_k = g_{k+1} - g_k$.

Set $k := k+1$, $m_{k+1} = \min\{m(k)+1, M\}$, go to Step 1.

Remark

1. The objective of Step 2 is to keep $\{\alpha_k\}$ uniformly bounded. In fact, for all k,

$$0 < \min\{\delta, \varepsilon\} < \alpha_k < \max\left\{\frac{1}{\varepsilon}, \delta\right\}.$$

2. The algorithm cannot cycle infinitely between Step 3 and Step 4. Indeed, since $0 < \eta < 1$, $1 - \mathrm{e}^{-t_k} > 0$ and $\langle \gamma'(0), g_k \rangle < 0$, the acceptance rule (8) in Step 3 is satisfied for sufficiently small value of t.

3. The choice of α_{k+1} in Step 4 comes from [11].

3 Convergence Analysis

Throughout this section, we assume that $f: \mathbf{R}^n \to \mathbf{R}^1$ is bounded. Given $x_0 \in \mathbf{R}^n$, the algorithm generates a sequence $\{x_k\} \subset \mathbf{R}^n$. In our analysis, we denote the level set of f by

$$L(x_0) = \{ x \in \mathbf{R}^n \mid f(x) \leqslant f(x_0) \}.$$

Lemma 2 Let $\{x_k\} \in \mathbf{R}^n$ be a sequence generated by the algorithm. If there exists $\varepsilon > 0$, such that

$$\| g_k \| \geqslant \varepsilon \tag{9}$$

for all k large enough, then we have that $\lim\limits_{k\to\infty} f(x_k)$ exists, i. e.,

$$\lim_{k\to\infty} f(x_k) = f^* \tag{10}$$

and

$$\lim_{k\to\infty}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|=0. \tag{11}$$

Proof According to the acceptance rule in Step 3, we have

$$f(\boldsymbol{x}_{l(k)})-f(\boldsymbol{x}+\boldsymbol{\gamma}_k)\geqslant-\eta(1-\mathrm{e}^{-t_k})\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\gamma}'_k(0). \tag{12}$$

Taking into account that $m_{(k+1)}\leqslant m_{(k)}+1$, and $f(\boldsymbol{x}_{k+1})\leqslant f(\boldsymbol{x}_{l(k)})$, we have $f(\boldsymbol{x}_{l(k+1)})\leqslant f(\boldsymbol{x}_{l(k)})$. This means that the sequence $\{f(\boldsymbol{x}_{l(k)})\}$ is nonincreasing for all k, and therefore $\{f(\boldsymbol{x}_{l(k)})\}\subset L(\boldsymbol{x}_0)$ is convergent.

By using (8) and (6), for all $k>M$, we have

$$f(\boldsymbol{x}_{l(k)})=f(\boldsymbol{x}_{l(k)-1}+\boldsymbol{\gamma}_{(l(k)-1)(t_{l(k)-1})})\leqslant\max_{0\leqslant j\leqslant m(l(k)-1)}\{f(\boldsymbol{x}_{l(k)-j-1})\}-\eta(1-\mathrm{e}^{-t_{l(k)-1}})\|\boldsymbol{g}_{l(k)-1}\|^2. \tag{13}$$

As $\{f(\boldsymbol{x}_{l(k)})\}$ is convergent, we obtain from (13) that

$$\lim_{k\to\infty}(-\eta)(1-\mathrm{e}^{-t_{l(k)-1}})\|\boldsymbol{g}_{l(k)-1}\|^2=0. \tag{14}$$

If the conclusion of Lemma 1 is true, then there exists some $\varepsilon>0$, such that for k large enough,

$$\|\boldsymbol{g}_k\|\geqslant\varepsilon.$$

This implies that

$$\lim_{k\to\infty}(1-\mathrm{e}^{-t_{l(k)}})=0,$$

which means

$$\lim_{k\to\infty}t_{l(k)}=0.$$

So, we have

$$\lim_{k\to\infty}\|\boldsymbol{\gamma}_{l(k)-1}\|=0.$$

Hence it can be derived from the Theorem in [3] that

$$\lim_{k\to\infty}f(\boldsymbol{x}_{l(k)})=\lim_{k\to\infty}f(\boldsymbol{x}_k). \tag{15}$$

By the rule for accepting the step $\boldsymbol{\gamma}_k(t_k)$, we have

$$f(\boldsymbol{x}_{k+1})-f(\boldsymbol{x}_{l(k)})\leqslant\eta(1-\mathrm{e}^{-t_k})\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\gamma}'_k(0)=-\eta(1-\mathrm{e}^{-t_k})\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_k. \tag{16}$$

Combining (14), (15) and (16) yields that

$$\lim_{k\to\infty}(1-\mathrm{e}^{-t_k})=0,$$

and hence

$$\lim_{k\to\infty}t_k=0.$$

Therefore

$$\lim_{k\to\infty}\|\boldsymbol{\gamma}_k\|=0,$$

which establishes (11).

Theorem 1 Let $\{\boldsymbol{x}_k\}$ be the sequence generated by the algorithm. Then

$$\lim_{k\to\infty}\inf\|\boldsymbol{g}_k\|=0. \tag{17}$$

Furthermore, no limit of the sequence $\{\boldsymbol{x}_k\}$ is a local maximizer of f.

Proof If the conclusion of the theorem is not true, without loss of generality, we assume that there exists some $\varepsilon>0$ and a positive index K, such that $\|\boldsymbol{g}_k\|\geqslant\varepsilon$, $\forall k>K$. By Lemma 2, we have that $\lim_{k\to\infty}\|\boldsymbol{\gamma}_k(t_k)\|=0$, and $\lim_{k\to\infty}t_k=0$. The acceptance rule (8) means that, for large enough k,

$$f\left(\boldsymbol{x}_k+\boldsymbol{\gamma}_k\left(\frac{t_k}{\omega}\right)\right)-f(\boldsymbol{x}_k)\geqslant f\left(\boldsymbol{x}_k+\boldsymbol{\gamma}_k\left(\frac{t_k}{\omega}\right)\right)-f(\boldsymbol{x}_{l(k)})>-\eta(1-\mathrm{e}^{-\frac{t_k}{\omega}})\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_k, \tag{18}$$

where $\omega\in(0,1)$. For composite function $f\left(\boldsymbol{x}_k+\boldsymbol{\gamma}_k\left(\frac{t_k}{\omega}\right)\right)$, we have

$$f\left(\boldsymbol{x}_k+\boldsymbol{\gamma}_k\left(\frac{t_k}{\omega}\right)\right)=f(\boldsymbol{x}_k)+\frac{t_k}{\omega}\frac{\mathrm{d}f(\boldsymbol{x}_k+\boldsymbol{\gamma}_k(t))}{\mathrm{d}t}\bigg|_{t=0}+o\left(\frac{t_k}{\omega}\right)=f(\boldsymbol{x}_k)-\frac{t_k}{\omega}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\gamma}'_k(0)+o\left(\frac{t_k}{\omega}\right). \tag{19}$$

From (18) and (19), we have from $\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\gamma}'_k(0)\leqslant0$ and $1-\mathrm{e}^{-\frac{t_k}{\omega}}<\frac{t_k}{\omega}$ that

$$(1-\eta)\frac{t_k}{\omega}g_k^{\mathrm{T}}\gamma'_k(0)+o\left(\frac{t_k}{\omega}\right)\geq\left[\frac{t_k}{\omega}-\eta(1-e^{-\frac{t_k}{\omega}})\right]g_k^{\mathrm{T}}\gamma'_k(0)+o\left(\frac{t_k}{\omega}\right)\geq 0. \tag{20}$$

Dividing (20) by $\frac{t_k}{\omega}$ and using $1-\eta>0$ and $(g_k)^{\mathrm{T}}\gamma'_k(0)\leq 0$, we obtain

$$\lim_{k\to\infty}(g_k)^{\mathrm{T}}\gamma'_k(0)=0. \tag{21}$$

From (6), (21) means that when $t_k\to 0$ as $k\to\infty$,

$$-\lim_{k\to\infty}\|g_k\|^2=\lim_{k\to\infty}g_k^{\mathrm{T}}\gamma'_k(0)=0, \tag{22}$$

which means that (17) is true.

Similar to the last part of Theorem in [3], we also get no limit point x^* of $\{x_k\}$ is a local maximizer of f.

4 Numerical Experiments

Numerical experiments on the new gradient-path algorithm with the nonmonotonic technique are performed. We set

$$M=10,\quad \eta=0.4,\quad \omega=0.2.$$

We compare the new algorithm (NMG) with an algorithm proposed in [11] which is called GBB. The numerical results are shown in Table 1. We report the final value (f), the number of iterations (IT), CPU time in seconds (time). For most of problems, the new algorithm is competitive in the number of iterations and CPU time.

Table 1 Results for our algorithm NMG and GBB

Problem	n	f	Iter	time
TRIGONOMETRIC(NMG)	100	3.263 2e-009	60	0.046 0
TRIGONOMETRIC (GBB)	100	3.117 6e-006	67	0.078 0
TRIGONOMETRIC(NMG)	1 000	3.679 8e-008	72	0.437 0
TRIGONOMETRIC (GBB)	1 000	4.861 6e-009	74	0.516 0
TRIGONOMETRIC(NMG)	10 000	5.447 2e-009	152	10.422 0
TRIGONOMETRIC (GBB)	10 000	9.353 0e-009	203	12.296 0
EXTENDED POWELL (NMG)	100	2.013 5e-006	249	0.062 0
EXTENDED POWELL (GBB)	100	1.834 7e-006	339	0.188 0
EXTENDED POWELL (NMG)	1 000	4.723 3e-006	251	0.375 0
EXTENDED POWELL (GBB)	1 000	4.016 8e-006	464	0.453 0
EXTENDED POWELL (NMG)	10 000	1.984 7e-006	288	2.438 0
EXTENDED POWELL (GBB)	10 000	5.640 2e-006	641	2.390 0
PENALTY (NMG)	100	9.024 9e-004	9	0.016 0
PENALTY (GBB)	100	9.024 9e-004	25	0.016 0
PENALTY (NMG)	1 000	0.010 315	15	0.062 0
PENALTY (GBB)	1 000	0.009 745	45	0.063 0
PENALTY (NMG)	10 000	0.099 0	22	0.187 0
PENALTY (GBB)	10 000	0.099 0	74	0.422 0
EXTENDED ROSENBROCK (NMG)	100	4.567 9e-008	121	0.032 0
EXTENDED ROSENBROCK (GBB)	100	2.924 4e-009	71	0.047 0
EXTENDED ROSENBROCK (NMG)	1 000	0.017 9	123	0.078 0
EXTENDED ROSENBROCK (GBB)	1 000	3.327 6e-010	89	0.094 0
EXTENDED ROSENBROCK (NMG)	10 000	1.203 9e-010	128	0.500 0
EXTENDED ROSENBROCK (GBB)	10 000	5.058 4e-010	90	0.391 0

Acknowledgements We are grateful to Miao Weihua and Zhou Qunyan for their useful comments and remarks that helped improving the paper in progress. We also thank two referees for their helpful comments.

[References]

[1] Botsaris C A, Jacobson D H. A Newton-type curvilinear search method for optimization [J]. Journal of Mathematical Analysis and Applications, 1976, 54(1): 217-229.

[2] Barzilai J, Borwein J M. Two-point step size gradient method [J]. IMA Journal of Numerical Analysis, 1988, 8(1): 141-148.

[3] Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for Newton method [J]. SIAM Journal on Numerical Analysis, 1986, 23(4): 707-716.

[4] Sun W. Nonmonotone optimization methods: motivation and development [C] // 4th International Conference on Numerical Linear Algebra and Optimization. Guilin, 2003.

[5] Sun W. Nonmonotone trust region method for solving optimization problems [J]. Applied Mathematics and Computation, 2004, 156(1): 159-174.

[6] Sun W, Yuan Y. Optimization Theory and Methods: Nonlinear Programming [M]. New York: Springer, 2006.

[7] Sun W, Zhou Q. An unconstrained optimization method using nonmonotone second order Goldstein's line search [J]. Science in China Series A: Mathematics, 2007, 50(10): 1 389-1 400.

[8] Yang Y, Sun W. Adaptive conic trust-region method for nonlinear least squares problems [J]. Journal of Nanjing Normal University, 2007, 30(1): 13-21.

[9] Zhang Y, Sun W, Qi L. A Nonmonotone filter Barzilai-Borwein method for optimization [J]. Asia Pacific Journal of Operational Research, 2010, 27(1): 55-69.

[10] Zhu D T. A family improved secant methods via nonmonotone curvilinear paths technique for equality constrained optimization [J]. Journal of Computational and Applied Mathematics, 2001, 136(1/2): 73-97.

[11] Raydan M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem [J]. SIAM J OPTIM, 1997, 7(1): 26-33.

[责任编辑:丁　蓉]