Pure Mathematics 理论数学, 2011, 1, 46-50
http://dx.doi.org/10.12677/pm.2011.11010 Published Online April 2011 (http://www.hanspub.org/journal/pm/)

A New Projected Gradient Method for Bound Constrained Optimization

Shanzhou Niu, Yi Wang, Dandan Cui
School of Mathematics and Computer Science, Gannan Normal University, Ganzhou
Email: maszniu@163.com
Received: Mar. 14th, 2011; revised: Mar. 28th, 2011; accepted: Apr. 1st, 2011.

Abstract: The projected gradient method is very suitable to solve large-scale nonlinear programming due to the simplicity of its iteration and implement. In this paper, based on the quasi-Cauchy equation and diagonal updating, a new projected gradient method is proposed for bound constrained optimization. On the basis of nonmonotone line search, global convergence is established. The numerical results show that the new algorithm is promising.

Keywords: Bound Constrained Optimization; Projected Gradient Method; Nonmonotone Line Search; Global Convergence

边界约束优化问题一个新的投影梯度方法

牛善洲，王 义，崔丹丹
赣南师范学院数学与计算机科学学院，赣州
Email: maszniu@163.com
收稿日期：2011 年 3 月 14 日；修回日期：2011 年 3 月 28 日；录用日期：2011 年 4 月 1 日

摘 要：投影梯度法因其算法简单、易于实现，非常适合求解大规模优化问题。本文基于拟柯西方程和对角变换，构造了一个新的投影梯度算法。在非单调线搜索条件下，证明该方法具有全局收敛性。最后数值实验表明新方法是有效的。

关键词：边界约束优化；投影梯度方法；非单调线搜索；全局收敛性

1. 引言

设 $f: R^n \to R$ 是一个连续可微的函数。考虑如下的边界约束优化问题：

$$\begin{aligned} &\min f(x) \\ &s.t.\ x \in \Omega = \left\{ x \in R^n \mid l \le x \le u \right\} \end{aligned} \tag{1}$$

其中，$l = \left(l^1, l^2, \cdots, l^n\right)^T$，$u = \left(u^1, u^2, \cdots, u^n\right)^T$，$-\infty \le l^i \le u^i \le +\infty, i = 1,2,\cdots,n$。将 f 在点 x 处的梯度记为 $g(x) = \left(g^1(x), g^2(x), \cdots, g^n(x)\right)^T$。

定义 1.1 设点 $\bar{x} \in \Omega$，如果 $\bar{x}$ 满足

$$\begin{cases} \bar{x}^i = l^i & \Rightarrow g^i > 0, \\ l^i < \bar{x}^i < u^i & \Rightarrow g^i = 0, \\ \bar{x}^i = u^i & \Rightarrow g^i < 0. \end{cases} \tag{2}$$

则称点 $\bar{x}$ 为问题(1)的一个稳定点。

问题(1)是一类十分重要的约束优化问题，许多实际的优化问题都可以转化为问题(1)的形式。此外，问题(1)常常是求解一般约束优化问题的增广 Lagrange 和罚函数方法的一个子问题。因此，近年来许多学者对问题(1)做了大量的研究，提出了许多求解该问题的数值算法[1-4]。

投影梯度方法具有易于实现，以及适合求解大规模优化问题的优点。另一方面，为了保证迭代点的有效性，计算迭代点的投影一般都是十分费时的。此外，即使投影的计算十分简便，投影梯度算法也会跟无约束优化问题的梯度方法一样具有很慢的收敛速度。为了提高投影梯度方法的收敛速度，文献[5]提出了求解问题(1)的一个谱投影梯度方法，此方法是无约束优化问题的谱梯度方法的推广。谱梯度方法最初是在文献

[6]中提出的，此方法提高了梯度方法的收敛速度并且大大减少了计算量。因此，谱梯度方法被广泛应用于求解无约束和约束优化问题[7-9]。

文献[10]基于拟柯西方程与对角变换提出了求解无约束优化问题的一个单调的梯度方法，此方法的主要思想是：如果对角变换得到的新的对角矩阵非正定时，将前一步的正定对角矩阵作为新的正定对角矩阵。文献[11]提出了无约束优化问题的一个多元谱梯度方法，并且具有二次终止性。此外，该方法引入非单调线搜索后具有全局收敛性。基于多元谱梯度方法，文献[4]提出了边界约束优化问题的一个多元谱投影梯度方法。基于拟柯西方程与对角变换，本文提出了一个新的投影梯度方法，并且采用文献[12]中的非单调线搜索技术和一些限制保证算法的全局收敛性。

2. 新的投影梯度方法

首先，考虑无约束优化问题的谱梯度方法。设 g_k 为函数 f 在点 x_k 处的梯度，谱梯度方法的迭代格式为

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k . \tag{3}$$

其中，α_k 由下述方式决定：

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}} . \tag{4}$$

其中，$s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1}$。步长 α_k 的选取可以减少计算量并且在很大程度上提高了梯度方法的收敛速度。此外，α_k 还被赋予了拟牛顿性质。事实上，由(4)式给出的 α_k 可以由下式得到：

$$\min_{\alpha} \left\| \alpha I s_{k-1} - y_{k-1} \right\|_2 ,$$

其中，$\alpha_k \mathrm{I}$ 近似代替函数 f 在点 x_k 处的 Hessian 矩阵。

我们在对角变换的基础上构造了一个新的梯度方法，其迭代格式如下：

$$x_{k+1} = x_k - H_k^{-1} g_k . \tag{5}$$

其中，矩阵 H_k 为对角矩阵。我们的目标是通过对角变换构造对角矩阵 H_k，使得矩阵 H_k 是 Hessian 矩阵的一个很好的近似。H_k 满足拟牛顿方程：

$$H_k s_{k-1} = y_{k-1} .$$

进一步，我们可以得到拟柯西方程：

$$s_{k-1}^T H_k s_{k-1} = s_{k-1}^T y_{k-1} .$$

设 $H_{k-1} > 0$ 是正定对角矩阵，H_k 是由对角变换得到的新矩阵并且使得 H_k 也是正定对角矩阵。为了使得 H_k 能够更好地近似 Hessian 矩阵，H_k 必须满足拟柯西方程，并且在变分原理下使得 H_k 和 H_{k-1} 的差量最小。在计算机中对角矩阵与向量占有相同的存储空间，因此我们可以得到一个存储空间为 $O(n)$ 的算法。文献[10]给出如下定理：

定理 2.1[10] 设 $\Lambda_{k-1} = H_k - H_{k-1}$，$s_{k-1} = x_k - x_{k-1}$，$y_{k-1} = g_k - g_{k-1}$。设 $s_{k-1} \neq 0$，H_{k-1} 是正定对角矩阵。考虑如下优化问题：

$$\begin{aligned} &\min \left\| \Lambda_{k-1} \right\|_F \\ &s.t.\ s_{k-1}^T \Lambda_{k-1} s_{k-1} = s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1} \end{aligned} . \tag{6}$$

其中，$\left\| \bullet \right\|_F$ 表示 F 范数。则(6)式的最优解为：

$$\Lambda_{k-1}^i = \frac{\left(s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1} \right) \left(s_k^i \right)^2}{tr\left(E^2 \right)}, i = 1, 2, \cdots, n$$

其中，tr 表示迹算子，Λ_k^i 是对角矩阵 Λ_k 的第 i 个对角元素，S_k^i 是 S_k 的第 i 个坐标元素，$E = \mathrm{diag}\left\{ \left(s_{k-1}^1 \right)^2, \left(s_{k-1}^2 \right)^2, \cdots, \left(s_{k-1}^n \right)^2 \right\}$。

由定理 2.1，我们得到 H_k 的迭代公式为：

$$H_k^i = H_{k-1}^i + \frac{(s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1}) \left(s_{k-1}^i \right)^2}{tr(E^2)}, i = 1, 2, \cdots, n . \tag{7}$$

其中，H_k^i，H_{k-1}^i 分别为 H_k，H_{k-1} 的第 i 个对角元素。若存在某个 i 使得 $H_k^i < 0$，则无法保证(5)式中的搜索方向为下降方向。为了克服上述缺点，我们使用下述的迭代格式：

$$x_{k+1} = x_k - \mathrm{diag}\left\{ \frac{1}{\lambda_k^1}, \frac{1}{\lambda_k^2}, \cdots, \frac{1}{\lambda_k^n} \right\} g_k . \tag{8}$$

其中，λ_k^i 由下式得到：

$$\lambda_k^i = \begin{cases} H_k^i, s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1} > 0, i = 1, 2, \cdots, n \\ \dfrac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1} \le 0, i = 1, 2, \cdots, n \end{cases} . \tag{9}$$

由(8)和(9)式，我们可以建立问题(1)的新的投影梯度算法。给定 $z \in R^n$，定义集合 Ω 上的投影 $p(z)$：

$$p(z) = \begin{cases} l^i, if\ z^i \le l^i, \\ z^i, if\ l^i < z^i < u^i, \\ u^i, if\ z^i \ge u^i. \end{cases} \tag{10}$$

为了保证算法的全局收敛性，我们使用文献[12]中的非单调线搜索技术和一些限制。下面我们给出新的投影梯度方法。

新的投影梯度(NPG)算法

给定数据：$x_0 \in R^n$，$\alpha_0 \in R$，初始矩阵 $H_0 = I$，$\gamma \in (0,1)$，$\delta > 0$，$0 < \sigma_1 < \sigma_2 < 1$，$\varepsilon > 0$，$\varepsilon_1 > 0$ 以及整数 $M \ge 1$。Set $k = 0$。

Step1：If $\left\| p\left(x_k - g_k\right) - x_k \right\| < \varepsilon_1$, stop.

Step2：If $k = 0$ then

Set $x_1 = p(x_0 - \alpha_0 g_0)$，跳转至 Step6。

If $s_{k-1}^T y_{k-1} - s_{k-1}^T H_{k-1} s_{k-1} > 0$，Set $\lambda_k^i = H_k^i$，$i = 1,2,\cdots,n$.

Otherwise $\lambda_k^i = \dfrac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, i = 1,2,\cdots,n.$

If $\lambda_k^i < \varepsilon$ or $\lambda_k^i > \dfrac{1}{\varepsilon}$，Set $\lambda_k^i = \delta.$

Step3：Set $\alpha_k = \operatorname{diag}\left\{ \dfrac{1}{\lambda_k^1}, \dfrac{1}{\lambda_k^2}, \cdots, \dfrac{1}{\lambda_k^n} \right\}$。

Step4：计算 $d_k = p\left(x_k - \alpha_k g_k\right) - x_k$；Set $\tau = 1$。

Step5：(非单调线搜索)

If $f\left(x_k + \tau d_k\right) \le \max\limits_{0 \le j \le \min(k, M-1)} f\left(x_{k-j}\right) + \gamma \tau g_k^T d_k$ then

令 $\tau_k = \tau, x_{k+1} = x_k + \tau_k d_k$，跳转至 Step6。

Else 取 $\tau_{new} = \left[\sigma_1 \tau, \sigma_2 \tau\right]$，Set $\tau = \tau_{new}$，跳转至 Step5。

Step6：Set $k = k + 1$；跳转至 Step1。

3. 全局收敛性分析

为了讨论 NPG 算法的全局收敛性，设目标函数 $f(x)$ 满足如下的基本假设：

假设 3.1 水平集 $\ell = \left\{ x \in \Omega \mid f(x) \le f(x_0) \right\}$ 是紧集。

引理 3.1 设 $x_k \in \Omega$，$d_k = p\left(x_k - \alpha_k g_k\right) - x_k$。则存在常数 $c_1 > 0$，使得 $g_k^T d_k \le -c_1 \|d_k\|^2$。

证明 由 d_k 的定义可知 $d_k^i = p\left(x_k^i - \dfrac{g_k^i}{\lambda_k^i} \right) - x_k^i$。下面分三种情况讨论：

Case 1：$x_k^i - \dfrac{g_k^i}{\lambda_k^i} \in \left(l^i, u^i\right)$，有

$$d_k^i = p\left(x_k^i - \frac{g_k^i}{\lambda_k^i} \right) - x_k^i = -\frac{g_k^i}{\lambda_k^i}，\text{则 } g_k^i d_k^i = -\lambda_k^i (d_k^i)^2 。$$

Case 2：$x_k^i - \dfrac{g_k^i}{\lambda_k^i} \le l^i$，有

$$d_k^i = p\left(x_k^i - \frac{g_k^i}{\lambda_k^i} \right) - x_k^i = l^i - x_k^i \le 0，\quad g_k^i \ge -\lambda_k^i \left(l^i - x_k^i\right)$$

则 $g_k^i d_k^i \le -\lambda_k^i \left(l^i - x_k^i\right) d_k^i = -\lambda_k^i (d_k^i)^2$。

Case 3：$x_k^i - \dfrac{g_k^i}{\lambda_k^i} \ge u^i$，有

$$d_k^i = p\left(x_k^i - \frac{g_k^i}{\lambda_k^i} \right) - x_k^i = u^i - x_k^i \ge 0，\quad g_k^i \le -\lambda_k^i \left(u^i - x_k^i\right)$$

则 $g_k^i d_k^i \le -\lambda_k^i \left(u^i - x_k^i\right) d_k^i = -\lambda_k^i (d_k^i)^2$。

由于 $\lambda_k^i \ge \varepsilon$，取 $c_1 = \varepsilon$，则 $g_k^T d_k \le -c_1 \|d_k\|^2$，引理得证。

引理 3.2 设 $\{x_k\}$ 是由 NPG 算法产生的点列。则 $d_k = 0 \Leftrightarrow x_k$ 是问题(1)的稳定点。

证明 定义 $L^k = \left\{ i \mid x_k^i = l^i \right\}$，$F^k = \left\{ i \mid l^i < x_k^i < u^i \right\}$，$U^k = \left\{ i \mid x_k^i = u^i \right\}$。

设 $d_k = 0$，如果 $i \in L^k$，$0 = d_k^i = p\left(x_k^i - \dfrac{g_k^i}{\lambda_k^i} \right) - x_k^i$ 则

$$p\left(x_k^i - \frac{g_k^i}{\lambda_k^i} \right) = x_k^i = l^i，$$

由(10)式得到 $x_k^i - \dfrac{g_k^i}{\lambda_k^i} \le l^i$。

考虑到 $\lambda_k^i > 0$，则得到 $g_k^i \ge 0$。

类似地，如果 $i \in U^k$，可以得到 $g_k^i \le 0$；如果 $i \in F^k$，可以得到 $g_k^i = 0$。因此，x_k 是问题(1)的稳定点。

另一方面，设 x_k 是问题(1)的稳定点，如果 $i \in L^k$，$g_k^i \ge 0$，则 $x_k^i - \dfrac{g_k^i}{\lambda_k^i} \le l^i$。因此，当 $i \in L^k$，得到 $d_k^i = 0$。类似地，当 $i \in F^k$ 可以得到 $d_k^i = 0$；$i \in U^k$ 可以得到 $d_k^i = 0$。因此，$d_k = 0$。引理得证。

由引理 3.1，引理 3.2 以及文献[5]中的定理 2.2，我们可以得到下面的收敛性定理。

定理 3.2 设 $\{x_k\}$ 是由 NPG 算法产生的点列，则 $\{x_k\}$ 的任一极限点都是问题(1)的稳定点。

4. 数值实验

在本节，我们对NPG算法用MATLAB编程，并在PC(Intel Core 2 Duo CPU 2.2GHz Memory 2GB)上进行数值实验。设置参数 $M = 5$，$\gamma = 10^{-4}$，$\sigma_1 = 0.1$，

1. $f(x)=\sum_{i=1}^{n}(e^{x_i}-x_i);\quad x_0=\left(\frac{1}{n},\frac{2}{n},\cdots,\frac{n}{n}\right)^T, l=(-100,-100,\cdots,-100)^T, u=(100,100,\cdots,100)^T$

Table 4.1.
表 4.1.

n	NI/NF/NG	CPU*(s)*	$\lVert p(x_k-g_k)-x_k\rVert$	$f(x_k)$
100	7/7/7	0.156	2.066143E–010	100
500	7/7/7	0.063	1.828211E–009	500
1000	7/7/7	0.125	3.526545E–009	1000
10000	7/7/7	11.297	1.838901E–008	10000

2. $f(x)=\sum_{i=1}^{n}\frac{i}{10}(e^{x_i}-x_i);\quad x_0=(1,1,\cdots,1)^T, l=(-1000,-1000,\cdots,-1000)^T, u=(1000,1000,\cdots,1000)^T$

Table 4.2.
表 4.2.

n	NI/NF/NG	CPU(s)	$\lVert p(x_k-g_k)-x_k\rVert$	$f(x_k)$
100	359/525/359	0.25	9.409498E–007	505
1000	268/339/268	4.25	9.717239E–007	50 050

3. $f(x)=\frac{1}{2}x^TAx;\quad A=\begin{pmatrix} n & & \\ & \ddots & \\ & & n \end{pmatrix}_{n\times n}\quad x_0=(1,1,\cdots,1)^T, l=(-10,-10,\cdots,-10)^T, u=(10,10,\cdots,10)^T$

Table 4.3.
表 4.3.

n	NI/NF/NG	CPU(s)	$\lVert p(x_k-g_k)-x_k\rVert$	$f(x_k)$
100	2/2/2	0.015	2.359224E–013	2.782969E–028
500	2/2/2	0.031	8.192363E–011	6.711481E–024
1000	2/2/2	0.094	1.025163E–009	5.254800E–022
5000	2/2/2	2.171	3.240671E–007	1.050195E–017

4. $f(x)=\frac{1}{2}x^TAx;\quad A=\begin{pmatrix} 1 & & \\ & \ddots & \\ & & n \end{pmatrix}_{n\times n}\quad x_0=(1,1,\cdots,1)^T, l=(-10,-10,\cdots,-10)^T, u=(10,10,\cdots,10)^T$

Table 4.4.
表 4.4.

n	NI/NF/NG	CPU(s)	$\lVert p(x_k-g_k)-x_k\rVert$	$f(x_k)$
100	69/73/69	0.11	7.468630E–007	1.100490E–013
200	120/151/120	0.157	9.889196E–007	1.871602E–013
300	90/100/90	0.312	7.179043E–007	8.276060E–014
500	361/516/361	3.641	9.324383E–007	1.409143E–013

$\sigma_2 = 0.9$， $\varepsilon = 10^{-10}$， $\alpha_0 = \dfrac{1}{\left\| p\left(x_0 - g_0\right) - x_0 \right\|}$，

$$\delta = \begin{cases} 1 & \text{if } \left\| p\left(x_k - g_k\right) - x_k \right\| > 1 \\ \left\| p\left(x_k - g_k\right) - x_k \right\|^{-1} & \text{if } 10^{-5} \le \left\| p\left(x_k - g_k\right) - x_k \right\| \le 1 \\ 10^5 & \text{if } \left\| p\left(x_k - g_k\right) - x_k \right\| < 10^{-5} \end{cases}。$$

我们对下述四个测试函数进行数值实验，迭代终止条件为：

$$\left\| p\left(x_k - g_k\right) - x_k \right\| \le 10^{-6}，即 \varepsilon_1 = 10^{-6}。$$

数值结果如表 4.1、表 4.2、表 4.3、表 4.4 所示。其中，n, NI, NF, NG, CPU 分别表示测试函数的维数，算法的迭代次数，函数值的计算次数，梯度值的计算次数，所用的 cpu 时间。

5. 致谢

本文第一作者感谢赣南师范学院研究生创新基金的资助。作者对审稿人和编辑所提出的宝贵意见表示衷心的感谢！

参考文献 (References)

[1] R. H. Byrd, P. Lu, J. Nocedal. A limited-memory algorithm for bound constrained optimization. SIAM J. Sci. Stat. Comput, 1995, 16(5): 1190-1208.

[2] W. W. Hager, H. Zhang. A new active set algorithm for box constrained optimization. SIAM J. Optim, 2006, 17(2): 526-557.

[3] J. Moré, G. Toraldo. On the solution of large scale quadratic programming problem with bound constrains. SIAM J. Optim, 1991, 1(1): 93-113.

[4] Z. S. Yu, Sun J, Qin Y. A multivariate spectral projected gradient method for bound constrained optimization. J. Comput. Appl. Math, 2011, 235(8): 2263-2269.

[5] E. G. Birgin, G. M. Martinez, M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim, 2000, 10(4): 1196-1211.

[6] J. Barzilai, J. M. Borwein. Two-point step size gradient methods. IMA J. Numer. Anal, 1988, 8(1): 141-148.

[7] R. Andreani, E. G. Birgin, J. M. Martínez, et al. Spectral projected gradient and variable metric methods for optimization with linear inequalities. IMA J. Numer. Anal, 2005, 25(2): 221-252.

[8] E. G. Birgin, G. M. Marttínez. Larges-scale active-set box-constrained optimization method with spectral projected gradients. Comput. Optim Appl, 2002, 23(1): 101-125.

[9] Y. Dai, R. Fletcher. Projected Barzilai-Borwein method for large-scale box-constrained quadratic programming. Numer. Math, 2005, 100(1): 21-47.

[10] W. J. Leong, M. A. Hassan, M. Farid. A monotone gradient method via weak secant equation for unconstrained optimization. Taiwanese J. Math, 2010, 14(2): 413- 423.

[11] L. Han, G. Yu, L. Guan. Multivariate spectral gradient method for unconstrained optimization. Appl Math Comput, 2008, 201(1-2): 621-630.

[12] L. Grippo, F. Lampariello, S. Licidis. A nonmonotone line search technique for Newtons method. SIAM Journal on Numerical Analysis, 1986, 23(4): 26-33.