文章编号:1007-6735(2009)02-0108-05

一类新的自适应非单调谱投影梯度法

林 骥, 宇振盛
(上海理工大学 理学院, 上海 200093)

摘要: 给出了求解凸约束优化的一类新的自适应非单调谱投影梯度法. 通过引入具有自适应性的权重参数,使算法在迭代过程中能自动调节非单调策略. 在适当条件下证明了算法的收敛性. 数值试验结果表明,该算法在一定程度上能减少在线搜索过程中对非单调参数 M 的依赖.

关键词: 约束优化; 非单调线搜索; 谱投影梯度法; 自适应算法
中图分类号: O 221.2 **文献标志码**: A

New class of adaptive nonmonotone spectral projected gradient method

LIN Ji, YU Zhen-sheng
(*College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China*)

Abstract: A new class of adaptive nonmonotone spectral projected gradient technique for convex constrained optimization was introduced. The main idea of the method is that a self-adaptive weight parameter which makes the algorithm be able to adjust the nonmonotone strategy automatically. The global convergence of the proposed algorithm under certain conditions was proved and the numerical tests show the algorithm can reduce the dependence on the nonmonotone parameter to some extend.

Key words: *constrained optimization*; *nonmonotone line search*; *spectral projected gradient method*; *adaptive algorithm*

1 问题的提出

考虑凸约束优化问题

$$\left.\begin{aligned}\min\quad & f(x)\\ \text{s.t.}\quad & x\in\Omega\end{aligned}\right\} \tag{1}$$

其中,$x\in\mathbb{R}^n$,$f:\mathbb{R}^n\to\mathbb{R}$,$f\in C^1$,$\Omega$ 为 $\mathbb{R}^n$ 中的闭凸集.

文献[1]利用两种谱投影梯度法(SPG1 和 SPG2,合称为 SPG)求解了问题(1),并做了大量的数值试验. 当 Ω 上的投影容易计算时,如带边界约束的最优化问题,SPG 算法在计算机上很容易编程实现,适合于求解大规模优化问题. SPG 算法结合了非单调线搜索技术[2]、投影梯度法[3]和两点步长梯度法[4].

文献[4]讨论了两点步长梯度法对其中非单调参数 M 的依赖问题. 在此基础上,文献[5]针对无约束优化问题提出了自适应两点步长梯度法,该算法在保证全局收敛性的前提下,让第一次的试验步

收稿日期: 2007-11-21
基金项目: 国家自然科学基金资助项目(10571106; 10671126)
作者简介: 林 骥(1982-),男,硕士研究生. E-mail: linji@live.com

长更容易被接受,因为,通常第一次的试验步长比第二次的候选步长含有更多目标函数的二阶信息.

文献[6]指出了文献[2]中非单调线搜索技术的一些缺点,并提出了用函数值的"取平均"代替"求最大"的思想,从数值试验结果来看,文献[6]中的非单调技术整体上要优于文献[2]. 更进一步,与采用单调线搜索方法相比,采用文献[2]中的非单调技术有时能够显著减少迭代次数,但有时也可能增加迭代次数[2,4-6]. 本文设计了一类新的自适应算法,称之为自适应非单调谱投影梯度法.与自适应两点步长梯度法[5]不同,该算法能够根据当前迭代点和前一次迭代点的函数值信息自动调节算法的单调策略,并且可以很方便地在单调与非单调之间进行有效的控制.数值试验结果表明,该新算法能够显著提高求解优化问题的效率.

此外,算法 SPG2 的数值表现有时非常依赖于参数 M 值的选择,M 值的细小变化有可能使算法 SPG2 的迭代次数发生显著的变化(参见后面的例2);相比之下,M 值在小范围内变化时,自适应非单调谱投影梯度法的数值表现相对比较稳定.

为方便起见,记 $g(x)=\nabla f(x)$, $f_k=f(x_k)$, $g_k=g(x_k)$.

2　自适应非单调谱投影梯度算法

先引入几个定义.

a. y 到 Ω 的投影

$$P_\Omega(y)=\arg\min_x\{\|x-y\|\mid x\in\Omega\},\quad y\in\mathbb{R}^n$$

b. 缩放了的投影梯度

$$g_\lambda(x)=P_\Omega[x-\lambda g(x)]-x,\ x\in\Omega,\lambda>0$$

c. 权重参数

$$\omega_k=\begin{cases}\left|\dfrac{1+|\min(f_{k-1},f_k)|}{1+|\max(f_{k-1},f_k)|}\right|^\delta, & k>0\\ 0, & k=0\end{cases},\delta>0$$

计算权重参数的主要思想是:若当前迭代点与前一迭代点的函数值相差较大,则 ω_k 接近于0,算法侧重于非单调策略;反之,若当前迭代点与前一迭代点的函数值相差较小,则 ω_k 接近于1,算法侧重于单调策略.其中,分子和分母同时加1的目的是保证算法在迭代过程中不会出现分母等于零的情形.

现介绍自适应非单调谱投影梯度算法(ANSPG).

a. 给定 $x_0\in\mathbb{R}^n$,若 $x_0\notin\Omega$,以 $P_\Omega(x_0)$ 替换 x_0. 取整数 $M\geqslant1$, $\gamma\in(0,1)$, $\delta\geqslant0$, $0<\mu_{\min}<\mu_0<\mu_{\max}<\infty$,令 $k=0$.

b. 如果 $\|P_\Omega(x_k-g_k)-x_k\|=0$,则算法终止.

c. 计算 $d_k=P_\Omega(x_k-\mu_k g_k)-x_k$,令 $t=1$.

d. 检验

$$f(x_k+td_k)\leqslant\omega_k f_k+(1-\omega_k)\cdot\max\{f_k,f_{k-1},\cdots,f_j\}+\gamma t\langle g_k,d_k\rangle \quad (2)$$

其中,$j=\max\{0,k-M+1\}$.

e. 如果式(2)不成立,则计算 $t_{\text{new}}\in[0.1t,0.9t]$,令 $t=t_{\text{new}}$;若 $t_{\text{new}}\notin[0.1t,0.9t]$,则令 $t=0.5t$. 转至 d.

f. 令 $t_k=t$, $x_{k+1}=x_k+t_kd_k$, $s_k=x_{k+1}-x_k$, $y_k=g_{k+1}-g_k$. 如果 $\langle s_k,y_k\rangle\leqslant0$,则令 $\mu_{k+1}=\mu_{\max}$;否则

$$\mu_{k+1}=\min\{\mu_{\max},\max\{\mu_{\min},\langle s_k,s_k\rangle/\langle s_k,y_k\rangle\}\}$$

$k=k+1$,转至 b.

在 d 中算法的单调策略可以通过 δ 的大小进行有效的控制. 当 $\delta=+\infty$ 时,$\omega_k\equiv0$,此时算法 ANSPG 即为文献[1]中的算法 SPG2;当 $\delta=0$ 时,$\omega_k\equiv1(k>0)$,则算法 ANSPG 即为 Armijo 单调线搜索方法(MSPG).

在 e 中可以采用与文献[1]相同的方法计算 t_{new},即采用二次插值法.设

$$\phi(t)=f(x_k+td_k)$$

开始时,有

$$\phi(0)=f(x_k),\quad \phi'(0)=\nabla f(x_k)^{\mathrm{T}}d_k \quad (3)$$

在计算了 $f(x_k+td_k)$ 以后,有

$$\phi(1)=f(x_k+d_k) \quad (4)$$

如果 $f(x_k+d_k)$ 不满足式(2),则用如下二次模型近似 $\phi(t)$.

$$m(t)=[\phi(1)-\phi(0)-\phi'(0)]t^2+\phi'(0)t+\phi(0)$$

上式满足式(3)和式(4)中的3个条件.令 $m'(t)=0$,得

$$\hat{t}=-\frac{\phi'(0)}{2[\phi(1)-\phi(0)-\phi'(0)]},\quad \text{令 } t_{\text{new}}=\hat{t}$$

另外,也可以采用与文献[2]相同的方法计算 t_{new},即只要式(2)不成立,就取 $t_{\text{new}}=0.5t$.

3　收敛性分析

记 $V_k=\omega_k f_k+(1-\omega_k)\max\{f_k,f_{k-1},\cdots,f_j\}$,其中,$j=\max\{0,k-M+1\}$. 从算法的 d 中 ω_k 的计算可知,$0\leqslant\omega_k\leqslant1$,故

$$f_k\leqslant V_k\leqslant\max\{f_k,f_{k-1},\cdots,f_j\} \quad (5)$$

引理1　对任意 $x\in\Omega$, $\lambda\in(0,\mu_{\max}]$

a. $\langle g(x), g_\lambda(x)\rangle \leqslant -\|g_\lambda(x)\|^2/\lambda \leqslant -\|g_\lambda(x)\|^2/\mu_{\max}$.

b. $g_\lambda(x^*)=0 \Leftrightarrow x^*$ 为受约束稳定点，即对任意 $x\in\Omega$，$\langle g(x^*), x-x^*\rangle \geqslant 0$.

其证明可参见文献[7].

定理1 设水平集 $L=\{x\in\mathbb{R}^n \mid f(x)\leqslant f(x_0)\}\cap\Omega$ 为非空有界紧集，则算法ANSPG是可行的，且迭代点列$\{x_k\}$的任一聚点 x^* 都是问题(1)的受约束稳定点.

证明 利用反证法. 若 x_k 不是受约束稳定点，由引理1，有

$$\langle g_k, d_k\rangle = \langle g_k, g_{\mu_k}(x_k)\rangle \leqslant -\|g_{\mu_k}(x_k)\|^2/\mu_{\max} < 0 \qquad (6)$$

即 d_k 是下降方向，算法的内循环 d—e—d 必定在有限步之内终止，所以，算法ANSPG是可行的.

设 $x^*\in\Omega$ 是迭代点列 $\{x_k\}$ 的聚点，则存在 $\{x_k\}$ 的一个子列 $\{x_{k_j}\}$ 收敛于 x^*，对子列重新编号，并将它仍然记为 $\{x_k\}$. 类似于文献[1]中定理2.4的证明，讨论以下两种情形.

情形1 $\inf\{t_k\}=0$.

假设 x^* 不是受约束稳定点. 由连续性，存在 $\xi>0$，使得对任意 $\mu\in[\mu_{\min},\mu_{\max}]$，有

$$\left\langle g(x^*), \frac{P_\Omega[x^*-\mu g(x^*)]-x^*}{\|P_\Omega[x^*-\mu g(x^*)]-x^*\|}\right\rangle < -\xi$$

由于 $x_k\to x^*$，所以，当 k 足够大时，对任意 $\mu\in[\mu_{\min},\mu_{\max}]$，有

$$\left\langle g_k, \frac{P_\Omega(x_k-\mu_k g_k)-x_k}{\|P_\Omega(x_k-\mu_k g_k)-x_k\|}\right\rangle < -\xi/2 \qquad (7)$$

另一方面，因为，$\inf\{t_k\}=0$，所以，存在无穷指标集 I，使得 $\lim\limits_{k\in I,k\to\infty} t_k=0$. 从算法ANSPG中 t_k 的计算方法可知，对于足够大的 $k\in I$，存在 $\sigma_k\in[0.1, 0.9]$，使得

$$f\left(x_k+\frac{t_k}{\sigma_k}d_k\right) > V_k+\gamma\frac{t_k}{\sigma_k}\langle g_k,d_k\rangle \geqslant f_k+\gamma\frac{t_k}{\sigma_k}\langle g_k,d_k\rangle$$

因此

$$\frac{f(x_k+\frac{t_k}{\sigma_k}d_k)-f_k}{t_k/\sigma_k} > \gamma\langle g_k,d_k\rangle$$

利用中值定理，有

$$\langle g(x_k+\theta_k d_k), d_k\rangle > \gamma\langle g_k,d_k\rangle$$

其中，$\theta_k\in[0,t_k/\sigma_k]$，$\lim\limits_{k\in I,k\to\infty}\theta_k=0$. 取一个子列使得 $d_k/\|d_k\|$ 收敛于 d，对上式两端取极限，得$(1-\gamma)\langle g(x^*),d\rangle\geqslant 0$. 由$(1-\gamma)>0$和$\langle g_k,d_k\rangle<0$，知$\langle g(x^*),d\rangle=0$. 利用连续性和 d_k 的定义，对于子列的足够大的 k，有

$$\left\langle g_k, \frac{P_\Omega(x_k-\mu_k g_k)-x_k}{\|P_\Omega(x_k-\mu_k g_k)-x_k\|}\right\rangle > -\xi/2$$

这与式(7)相矛盾.

情形2 $\inf\{t_k\}\geqslant\rho>0$.

根据引理1的b，若 x^* 不是受约束稳定点，则对任意 $t\in(0,\mu_{\max}]$，有

$$\|P_\Omega[x^*-tg(x^*)]-x^*\|>0$$

由连续性可知，存在 $\xi>0$，使得对任意 $t\in[\rho,\mu_{\max}]$，有

$$\|P_\Omega[x^*-tg(x^*)]-x^*\|>\xi$$

由于 $x_k\to x^*$ 及 $\inf\{t_k\}\geqslant\rho>0$，故当 k 足够大时，有

$$\|P_\Omega[x_k-t_k g(x_k)]-x_k\|>\xi/2$$

设整数 $l(k)$ 满足

$$f_{l(k)}=\max\{f_k,f_{k-1},\cdots,f_{k-j}\}$$

其中，$j=\max\{0,k-M+1\}$，$k-j\leqslant l(k)\leqslant k$.

根据式(6)，$\langle g_k,d_k\rangle<0$. 另外，在算法ANSPG的运行过程中，有 $\gamma>0$，$t>0$. 故 $\gamma t\langle g_k,d_k\rangle<0$. 所以，由式(1)，有

$$f(x_k+td_k)\leqslant\omega_k f_k+(1-\omega_k)\max\{f_k,f_{k-1},\cdots,f_j\}$$

又因为

$$V_k=\omega_k f_k+(1-\omega_k)\max\{f_k,f_{k-1},\cdots,f_j\}$$

$f_k\leqslant\max\{f_k,f_{k-1},\cdots,f_j\}$，以及 $0\leqslant\omega_k\leqslant1$，所以，$V_k\leqslant\max\{f_k,f_{k-1},\cdots,f_j\}$，即 $V_k\leqslant f_{l(k)}$. 从而有

$$f(x_k+td_k)\leqslant V_k\leqslant f_{l(k)}$$

从算法ANSPG的f中可知，$t_k=t$，$x_{k+1}=x_k+t_k d_k$，故 $f_{k+1}=f(x_{k+1})\leqslant f_{l(k)}$，从而 $\max\{f_{k+1},f_{l(k)}\}=f_{l(k)}$.

另一方面，由 $f_{l(k)}=\max\{f_k,f_{k-1},\cdots,f_{k-j}\}$ 可知，$f_{l(k+1)}=\max\{f_{k+1},f_k,\cdots,f_{k+1-j}\}$，根据

$$\max\{f_{k+1},f_k,\cdots,f_{k+1-j}\}\leqslant\max\{f_{k+1},f_k,\cdots,f_{k+1-j},f_{k-j}\}=\max\{f_{k+1},\max\{f_k,f_{k-1},\cdots,f_{k-j}$$

有

$$f_{l(k+1)}\leqslant\max\{f_{k+1},\max\{f_k,f_{k-1},\cdots,f_{k-j}\}\}=\max\{f_{k+1},f_{l(k)}\}=f_{l(k)}$$

所以，$\{f_{l(k)}\}$ 是单调非增的序列. 再利用引理1的a，有

$$f_{l(k)}\leqslant V_{l(k)-1}+\gamma t_{l(k)-1}\langle g_{l(k)-1},g_{\mu_{l(k)-1}}(x_{l(k)-1})\rangle\leqslant$$

$$f_{l(l(k)-1)}-\frac{\gamma\rho\|g_{\mu_{l(k)-1}}(x_{l(k)-1})\|^2}{\mu_{max}}\leqslant f_{l(l(k)-1)}-\frac{\gamma\rho\xi^2}{4\mu_{max}}$$

因此,$\lim\limits_{k\to\infty}f_{l(k)}=-\infty$.但由 $x_k\to x^*$ 及 f 是连续函数,可知 $\lim\limits_{k\to\infty}f_k=f(x^*)$.矛盾.

4　数值试验和比较

同文献[1]中的两个算法一样,本文给出的算法ANSPG也属于投影算法,适用于求解带边界约束的最优化问题,该类问题是利用增广 Lagrange 法求解一般非线性规划问题的必要工具[8-10].为此,本文针对边界约束最优化问题进行了数值试验.

在一台 CPU 2.93G,RAM 512M 的 PC 上利用 Matlab7.4.0 编程实现了算法 ANSPG,并与算法 SPG2 和 MSPG(单调谱投影梯度法)进行了数值比较.算法的初始值设置为

$\mu_0=1/\|P_\Omega(x_0-g_0)-x_0\|_\infty$,$\gamma=10^{-4}$

$\mu_{\min}=10^{-30}$,$\mu_{\max}=10^{30}$,$M=10$

在算法 ANSPG 的 e 中,采用与文献[2]相同的方法计算 t_{new},即若式(1)不成立,则 $t_{new}=0.5t$.

终止准则为

$$\|P_\Omega(x_k-g_k)-x_k\|_\infty<10^{-6}$$

例 1　对于严格凸函数[4]

$$f(x)=\sum_{i=1}^{n}i(e^{x_i}-x_i)/10$$

求解

$$\min\quad f(x)$$

$$\text{s.t.}\quad -10\leqslant x_i\leqslant 10,\quad i=1,2,\cdots,n$$

其中,变量数(维数) n 可以根据需要设置为任何整数,试验分别取 $n=6\,000,7\,000,8\,000,9\,000$ 和 $10\,000$.取 $x_0=(1,1,\cdots,1)^{\mathrm{T}}$.

该例只含有边界约束,根据投影的定义,投影的计算为

$$\forall x=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbb{R}^n$$

$$P_\Omega(x)=(\max\{-10,\min\{10,x_1\}\},\cdots,\max\{-10,\min\{10,x_n\}\})^{\mathrm{T}}$$

数值试验结果如表 1 所示.

表 1　算法 MSPG、SPG2 和 ANSPG 的数值试验结果

Tab.1　Numerical results of algorithms MSPG, SPG2 and ANSPG

n	MSPG($\delta=0$)			SPG2($\delta=\infty$)			ANSPG($\delta=100$)		
	I	N	T	I	N	T	I	N	T
6 000	1 277	2 981	5.01	1 574	2 361	5.79	1 100	2 396	4.27
7 000	1 434	3 204	6.26	1 671	2 573	6.96	672	1 576	3.03
8 000	1 160	2 458	5.52	1 901	2 866	8.71	1 269	2 909	6.17
9 000	656	1 568	3.72	2 001	3 058	10.15	955	2 094	5.23
10 000	887	2 160	5.56	1 948	2 972	11.08	1 144	2 693	7.03
求　和	5 414	12 371	26.07	9 095	13 830	42.69	5 140	11 668	25.73

在表 1 中,I 表示迭代次数,N 表示目标函数估值次数,T 表示 CPU 运行时间.因为,在线搜索过程中没有进行梯度估值计算,所以,梯度估值次数与迭代次数相同,在表 1 中没有单独列出.从表 1 中可以看出,对于本例来讲,$n=8\,000,9\,000$ 和 $10\,000$ 时,SPG2 明显比 MSPG 的数值表现差.但取 $\delta=100$ 时,ANSPG 的迭代次数与目标函数估值次数之和比 MSPG 少 5.49%,比 SPG2 少 26.68%;ANSPG 的总 CPU 运行时间比 MSPG 少 1.3%,比 SPG2 少 39.73%.

例 2　对于 Trigonometric 函数[11]

$$f(x)=1+\sum_{i=1}^{n}x_i+1\,000\Big(1-\sum_{i=1}^{n}\frac{1}{x_i}\Big)^2+1\,000\Big(1-\sum_{i=1}^{n}\frac{i}{x_i}\Big)^2$$

求解

$$\min\quad f(x)$$

$$\text{s.t.}\quad 0.01\leqslant x_i\leqslant 10\,000,\quad i=1,2,\cdots,n$$

取 $n=15$,$x_0=(1,1,\cdots,1)^{\mathrm{T}}$,$M=5,6,7,8,9$ 和 10.

类似于例 1,本例中投影的计算为

$$\forall x=(x_1,\cdots,x_n)^{\mathrm{T}}\in\mathbb{R}^n$$

$$P_\Omega(x)=(\max 0.01,\min\{10\,000,x_1\}\},\cdots,\max\{0.01,\min\{10\,000,x_n\}\})^{\mathrm{T}}$$

数值试验结果如表 2 所示(见下页).从表 2 中可以看出,本例中算法 SPG2 对 M 值的变化相对比较敏感,M 分别取 8,9 和 10 时,算法 SPG2 需要的总迭代次数分别为 701,1 822 和5 489;但对于算法 ANSPG 而言,M 值在小范围内变化对算法 ANSPG 的影响不大,取 $\delta=10$ 时所需的总迭代次数分别为

835,628和628,取$\delta=100$时所需的总迭代次数分别为1 118,898和898. 表2中最后一行计算了各列中除求和以外的数值的标准差,它们的大小也可以反应出不同算法对参数M值的敏感程度.

表2 算法SPG2与ANSPG对不同M值的数值试验结果

Tab.2 Numerical results of algorithms SPG2 and ANSPG with different M

M	SPG2($\delta=\infty$)			ANSPG($\delta=10$)			ANSPG($\delta=100$)		
	I	N	T	I	N	T	I	N	T
5	156	252	0.16	243	544	0.24	251	584	0.25
6	175	272	0.18	243	544	0.24	251	584	0.25
7	175	272	0.18	251	584	0.25	356	823	0.35
8	272	429	0.26	251	584	0.25	347	771	0.34
9	657	1 165	0.63	191	437	0.19	267	631	0.26
10	1 847	3 642	1.81	191	437	0.19	267	631	0.26
求 和	3 282	5 852	3.22	1 370	3130	1.36	1 739	4 024	1.71
标准差	664.4	1338.1	0.65	29.1	68.0	0.43	48.4	101.4	0.05

5 讨 论

文献[12]提出了两点步长梯度法,其基本思想是利用当前迭代点及前一迭代点的信息来确定步长因子.文献[13]对无约束优化问题提出了修正的两点步长梯度法,此算法在计算步长时利用了函数值以及梯度的信息,因此,数值试验的结果得到了改善.如果将修正的步长选择策略应用于本文的ANSPG算法,那么数值试验的结果有望得到进一步的改善. 另外,从表2中可以看出,算法ANSPG所需的迭代次数与δ的取值有关,因此,如何恰当地选取权重参数仍是值得进一步研究的问题.

参考文献:

[1] BIRGIN E G, MARTINEZ J M, RAYDAN M. Nonmonotone spectral projected gradient methods on convex sets[J]. SIAM Journal on Optimization, 2000, 10(4): 1 196-1 211.

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

[3] BERTSEKAS D P. On the goldstein-levitin-polyak gradient projection method[J]. IEEE Transactions on Automatic Control, 1976, 21(2): 174-184.

[4] RAYDAN M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem[J]. SIAM Journal on Optimization, 1997, 7(1): 26-33.

[5] DAI Y H, ZHANG H C. Adaptive two-point stepsize gradient algorithm[J]. Numerical Algorithms, 2001, 27(4): 377-385.

[6] ZHANG H C, HAGER W W. A nonmonotone line search technique and its application to unconstrained optimization[J]. SIAM Journal on Optimization, 2004, 14(4): 1 043-1 056.

[7] BERTSEKAS D P. Nonlinear Programming[M]. Belmont: Athena Scientific, 1995.

[8] CONN A R, GOULD N I M, TOINT P L. Global convergence of a class of trust region algorithms for optimization with simple bounds[J]. SIAM Journal on Numerical Analysis, 1989, 25(2): 764-767.

[9] CONN A R, GOULD N I M, TOINT P L. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds [J]. SIAM Journal on Numerical Analysis, 1991, 28(2): 545-572.

[10] FRIEDLANDER A, MARTINEZ J M, SANTOS S A. A new trust tegion algorithm for bound constrained minimization[J]. Applied Mathematics and Optimization, 1994, 30(3): 235-266.

[11] CHEN Z W, HAN J Y, XU D C. A nonmonotone trust region method for nonlinear programming with simple constraints[J]. Applied Mathematics and Optimization, 2001, 43(1): 63-85.

[12] BARZILAI J, BORWEIN J M. Two-point step size gradient methods[J]. IMA Journal of Numerical Analysis, 1988, 8(1): 141-148.

[13] DAI Y, YUAN J, YUAN Y X. Modified two-point stepsize gradient methods for unconstrained optimization [J]. Computational Optimization and Applications, 2002, 22(1): 103-109.