文章编号：1674-8085(2016)01-0015-07

一种求解无约束优化问题的修正共轭梯度算法

王晓亮，*袁功林，段侠彬

（广西大学数学与信息科学学院，广西，南宁 530004）

摘 要：提出了一种求解无约束优化问题修正的共轭梯度算法，该算法具有函数值信息，而且对线搜索技术具有加速作用，另外该算法不仅具有充分下降性还在适当条件下具有全局收敛性。数值结果也表明该算法对测试问题是有效的。

关键词：共轭梯度；函数值；加速项；充分下降

中图分类号：O224 **文献标识码**：A DOI:10.3969/j.issn.1674-8085.2016.01.004

MODIFIED CONJUGATE GRADIENT ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

WANG Xiao-liang, *YUAN Gong-lin, DUAN Xia-bin

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

Abstract: A modified conjugate gradient method with function information and acceleration scheme of line search for unconstraint optimization problems is presented. Furthermore, the proposed method not only possesses sufficient descent property but also has global convergence in mild conditions. The numerical results indicate that the presented method is effective for the test problems.

Key words: conjugate gradient; function information; acceleration scheme; sufficient descent

1 引言

考虑下面无约束问题：

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

其中 $f(x):\Re^n\to\Re$ 是一个连续可微的函数。共轭梯度法由于其简单性和低存储性成为求解(1.1)的最有效方法之一。共轭梯度法的迭代公式通常被定义如下：

$$x_{k+1}=x_k+\alpha_k d_k,\quad k=0,1,2,\cdots \tag{1.2}$$

其中 x_k 是第 k 个迭代点；α_k 是步长，它由一些线搜索确定；d_k 是搜索方向，它通常定义为：

$$d_k=\begin{cases}-g_k, & if\ \ k=0,\\ -g_k+\beta_k d_{k-1}, & if\ \ k\ge 1.\end{cases} \tag{1.3}$$

其中 g_k 是 $g(x_k)$ 的缩写，$g(x_k)$ 表示函数 $f(x)$ 在点 x_k 的梯度 $\nabla f(x_k)$；β_k 是一个标量。式(1.3)也表明不同的共轭梯度法具有不同的 β_k。

在过去的几十年里，不少学者提出了一些著名的共轭梯度法例 FR 方法[1]，HS 方法[2]和 WYL 方法[3-4]等。它们的 β_k 列出如下：

$$\beta_k^{FR}=\frac{\|g(x_{k+1})\|^2}{\|g(x_k)\|^2},\quad \beta_k^{HS}=\frac{g_{k+1}^T y_k}{d_k^T y_k},$$

收稿日期：2015-10-24；修改日期：2015-11-19

基金项目：国家自然科学基金项目(11261006);广西杰出青年科学基金项目(2015GXNSFGA139001)

作者简介：王晓亮(1989-)，男，河南平顶山人，硕士生，主要从事最优化方向研究(E-mail:xliangwang@126.com);

*袁功林(1976-)，男，河南商丘人，教授，博士生导师，主要从事最优化理论与方法等方向研究(E-mail:yuangl0417@126.com);

段侠彬(1990-)，男，湖南衡阳人，硕士生，主要从事最优化方向研究(E-mail:xiabinduan@126.com).

$$\beta_k^{WYL}=\frac{g_{k+1}^T\left(g_{k+1}-\frac{\|g_{k+1}\|}{\|g_k\|}g_{k+1}^Tg_k\right)}{\|g_k\|^2}.$$

在共轭梯度法的全局收敛性分析中，弱的 Wolfe-Powell 线搜索技术是非常有用的，它常见的表达形式为：

$$f\left(x_k+\alpha_kd_k\right)\le f\left(x_k\right)+\delta\alpha_kg_k^Td_k, \tag{1.4}$$

$$g\left(x_k+\alpha_kd_k\right)^Td_k\ge\sigma g\left(x_k\right)^Td_k. \tag{1.5}$$

其中$\delta\in\left(0,\frac{1}{2}\right),\sigma\in\left(\delta,1\right)$。该技术通常用来证明目标函数为强凸的非线性共轭梯度法的收敛性。

Zhang 和 Hager[5] 2005 年提出了一个新的非单调线搜索技术，即：

$$f\left(x_k+\alpha_kd_k\right)\le C_k+\delta\alpha_kg\left(x_k\right)^Td_k, \tag{1.6}$$

$$g\left(x_k+\alpha_kd_k\right)^Td_k\ge\sigma g\left(x_k\right)^Td_k, \tag{1.7}$$

$$Q_{k+1}=\eta_kQ_k+1,\ C_{k+1}=\left(\eta_kQ_kC_k+f\left(x_k+\alpha_kd_k\right)\right)/Q_{k+1}. \tag{1.8}$$

其中$0\le\eta_{\min}\le\eta_{\max}\le1$，$\eta_k\in\left[\eta_{\min},\eta_{\max}\right]$，$0<\delta\le\sigma<1$，$C_0=f\left(x_0\right)$，$Q_0=1$。

由于 $C_0=f\left(x_0\right)$，从而得到C_k是函数值$f\left(x_0\right),f\left(x_1\right),\cdots,f\left(x_k\right)$的一个凸组合。参数$\eta_k$控制着非单调的程度，即：如果对每个 k，都有$\eta_k=0$，则该线搜索转变为通常的单调 Wolfe 或 Armijo 线搜索；如果对每个 k，都有$\eta_k=1$，则C_k是函数值的平均数：若记$A_k=\frac{1}{k+1}\sum_{i=0}^kf\left(x_i\right)$，则有$C_k=A_k$。该技术通常用于证明目标函数为一般非线性时共轭梯度法的收敛性，同时数值结果也表明该方法比一般的非单调技术更有效。

下面的充分下降条件

$$g_k^Td_k\le-c\|g_k\|^2,\quad\forall\ k\ge0. \tag{1.9}$$

也常常用来分析非线性共轭梯度法的全局收敛性，其中 c 是一个正的常数。Gilbert 和 Nocedal[6]，Hu 与 Storey[7]指出充分下降性对共轭梯度法是至关重要的。一些共轭梯度方法[8-11]拥有与线搜索无关的充分下降性。

2004 年 Zhang 和 Hager [12]提出了一族共轭梯度方法：

$$\beta_k^\theta=\frac{g_{k+1}^Ty_k}{d_k^Ty_k}-\theta_k*\frac{g_{k+1}^Td_k\|y_k\|^2}{\left(d_k^Ty_k\right)^2}=\beta_k^{HS}-\theta_k*\frac{g_{k+1}^Td_k\|y_k\|^2}{\left(d_k^Ty_k\right)^2}$$

其中$y_k=g_{k+1}-g_k$，$\theta_k\ge0$。参数θ_k控制着共轭性和下降性的权重：当θ_k从右边趋近于 0 时，标量β_k^θ趋近于β_k^{HS}；当θ_k趋近于无穷大时，充分下降性里的参数 c 趋近于 1。该族共轭梯度方法对强凸的函数具有全局收敛性，但对一般的非线性函数并不具有，其原因是标量β_k^θ可能是负的。为了解决这个问题，Zhang 和 Hager[12]做了如下修改：

$$\beta_k^N=\max\left\{\beta_k^\theta,\omega_k\right\},\qquad\omega_k=-\frac{1}{\|d\|\min\left\{\omega,\|g_k\|\right\}}. \tag{1.10}$$

其中$\omega>0$是一个常数。

本文结构安排如下：在第二节里，新的算法将被提出，充分下降性和全局收敛性在第三节中得到证明；第四节将给出数值试验及其结果。

2 算法

逆牛顿法是解决无约束优化问题常用的方法，而 BFGS 方法又是逆牛顿法中最常用的方法，它的常见形式如下：

$$B_{k+1}=B_k-\frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k}+\frac{y_ky_k^T}{s_k^Ty_k}$$

其中B_k是目标函数 Hesse 矩阵的近似，$S_k=x_{k+1}-x_k$。然而在迭代过程中，没有用到函数值的信息，为此 Wei 等[13]提出了一个修正的 BFGS 迭代公式：

$$B_{k+1}=B_k-\frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k}+\frac{y_k^*y_k^{*T}}{s_k^Ty_k^*}$$

其中$y_k^*=y_k+\frac{\rho_k}{\|s_k\|^2}s_k$，$\rho_k=2\left(f_k-f_{k+1}\right)+\left(g_k+g_{k+1}\right)^Ts_k$。然而如果目标函数是一般凸函数时，修正的 BFGS 算法不满足超线性收敛。为了克

服该缺陷，Yuan 与 Wei[14]提出了新的修正公式：

$$B_{k+1}=B_k-\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}+\frac{y_k^m y_k^{mT}}{s_k^T y_k^m} \qquad (2.1)$$

其中 $y_k^m=y_k+\dfrac{\max\{\rho_k,0\}}{\|s_k\|^2}$。新的修正公式(2.1)能够确保对一般的凸函数，迭代项 B_k 是正定的，同时数值结果也表明该算法是有效的。

通过上述讨论，在方向 d_k 计算和迭代过程中，引入函数值信息是有效的，其迭代公式为：

$$d_k=\begin{cases}-g_k, & if \quad k=0,\\ -g_k+\beta_k^m d_{k-1}, & if \quad k\ge 1.\end{cases} \qquad (2.2)$$

其中 $\beta_k^m=\max\{\beta_k^{\theta m},\omega_k\}$，

$$\beta_k^{\theta m}=\frac{g_{k+1}^T y_k^m}{d_k^T y_k^m}-\frac{\theta_k g_{k+1}^T d_k\|y_k^m\|^2}{\left(d_k^T y_k^m\right)^2}$$

。

在共轭梯度算法中，通常会遇到这样的问题：搜索方向 d_k 往往比较困难去衡量，因此线搜索不得不计算更多的函数值以便找到合适的步长 α_k。为了克服这类问题，文献[15-16]提出了增加乘子修正步长 α_k 的方法，以便增加函数值沿着迭代方向的减少量，其迭代公式如下：

$$x_{k+1}=x_k+\bar{\alpha}_k d_k,\quad k=0,1,2,\cdots \qquad (2.3)$$

其中 $\bar{\alpha}_k=\gamma_k\alpha_k$，$\gamma_k=-\dfrac{\xi_k}{\lambda_k}$，$\xi_k=\alpha_k g_k^T d_k$，$\lambda_k=-\alpha_k(g_k-g_z)^T d_k$，$g_z=g(z)$，且 $z=x_k+\alpha_k d_k$。如果 $\lambda_k>0$，那么新的迭代点由公式(2.3)计算，否则由公式(1.2)计算。

基于上述讨论，现提出新的共轭梯度算法

Algorithm N

步 0： 给定初始点 $x_0\in\Re^n$, $0<\delta\le\sigma<1$, $\varepsilon>0$，令 $k:=0$；

步 1：若 $\|g_k\|\le\varepsilon$，则停止；否则进行下一步；

步 2：通过线搜索技术(1.6),(1.7),(1.8)计算步长 α_k；

步 3：通过公式(2.2),计算搜索方向 d_k；

步 4：加速项计算。计算 $z=x_k+\alpha_k d_k$，$\xi_k=\alpha_k g_k^T d_k$, $\lambda_k=-\alpha_k\left(g_k-g_z\right)^T d_k$；

步 5：若 $\lambda_k>0$, 下一个迭代点由公式(2.3)计算，否则由公式(1.2)计算；

步 6：若 $\|g_{k+1}\|\le\varepsilon$, 则停止；否则，令 $k:=k+1$, 返回第二步。

3 充分下降性和全局收敛性

与文献[8]中定理 1.1 类似，得到下面的引理 3.1。

引理 3.1 $\{d_k,\beta_k\}$ 由公式(2.2)给出，则对任意的 $\theta_k>\dfrac{1}{4}$ 和 $\beta\in\left[\beta_k^m,\max\{\beta_k^m,0\}\right]$，有

$$d_k^T g_k\le-\left(1-\frac{1}{4\theta_k}\right)\|g_k\|^2 \qquad (3.1)$$

证明：由 $d_0=-g_0$, 有

$$g_0^T d_0=-\|g_0\|^2\le-\left(1-\frac{1}{4\theta_k}\right)\|g_0\|^2$$

用 g_k^T 与 d_k 做内积，同时假设 $\beta=\beta_k^m$，那么可以得到下式：

$$g_k^T d_k=-\|g_k\|^2+\beta_k^m g_k^T d_{k-1}=-\|g_k\|^2+\frac{y_{k-1}^{mT}g_k}{d_{k-1}^T y_{k-1}^m}g_k^T d_{k-1}-\frac{\theta_k\|y_{k-1}^m\|^2\left(d_{k-1}^T g_k\right)^2}{\left(d_{k-1}^T y_{k-1}^m\right)^2}=$$

$$\frac{y_{k-1}^{mT}g_k g_k^T d_{k-1}d_{k-1}^T y_{k-1}^m-\|g_k\|^2\left(d_{k-1}^T y_{k-1}^m\right)^2-\theta_k\|y_{k-1}^m\|^2\left(d_{k-1}^T g_k\right)^2}{\left(d_{k-1}^T y_{k-1}^m\right)^2}=$$

$$\frac{y_{k-1}^{mT}g_k\left(d_{k-1}^T y_{k-1}^m\right)\left(g_k^T d_{k-1}\right)-\|g_k\|^2\left(d_{k-1}^T y_{k-1}^m\right)^2-\theta_k\|y_{k-1}^m\|^2\left(d_{k-1}^T g_k\right)^2}{\left(d_{k-1}^T y_{k-1}^m\right)^2}$$

令 $u=\dfrac{1}{\sqrt{2\theta_k}}\left(d_{k-1}^T y_{k-1}^m\right)g_k$, $v=\sqrt{2\theta_k}\left(g_k^T d_{k-1}\right)y_{k-1}^m$，则有：

$$g_k^T d_k = \frac{y_{k-1}^{mT} g_k \left(d_{k-1}^T y_{k-1}^m\right)\left(g_k^T d_{k-1}\right) - \|g_k\|^2 \left(d_{k-1}^T y_{k-1}^m\right)^2 - \theta_k \left\|y_{k-1}^m\right\|^2 \left(d_{k-1}^T g_k\right)^2}{\left(d_{k-1}^T y_{k-1}^m\right)^2} \le$$

$$\frac{\frac{1}{2}\left(\|u\|^2 + \|v\|^2\right) - \|g_k\|^2 \left(d_{k-1}^T y_{k-1}^m\right)^2 - \theta_k \left\|y_{k-1}^m\right\|^2 \left(d_{k-1}^T g_k\right)^2}{\left(d_{k-1}^T y_{k-1}^m\right)^2} = -\left(1 - \frac{1}{4\theta_k}\right)\|g_k\|^2 .$$

如果$\beta \ne \beta_k^m$,那么有$\beta_k^m < \beta \le 0$。通过g_k^T与d_k做内积，有

$$g_k^T d_k = -\|g_k\|^2 + \beta g_k^T d_{k-1}$$

若$g_k^T d_{k-1} \ge 0$，则

$$g_k^T d_k \le -\|g_k\|^2 \le -\left(1 - \frac{1}{4\theta_k}\right)\|g_k\|^2$$

若$g_k^T d_{k-1} < 0$,则：

$$g_k^T d_k = -\|g_k\|^2 + \beta g_k^T d_{k-1} \le -\|g_k\|^2 + \beta_k^m g_k^T d_{k-1}$$

由上面的讨论可知式(3.1)成立。故对任意的正整数，式(3.1)成立。

由 ω_k 的定义可知：$\omega_k < 0$。又$\beta_k^m = \max\left\{\beta_k^{\theta m}, \omega_k\right\} \in \left[\beta_k^m, \max\left\{\beta_k^m, 0\right\}\right]$，故对任意的正整数$k$和由 Algorithm N 产生的$\{d_k, g_k\}$，都有：

$$g_k^T d_k \le -\left(1 - \frac{1}{4\theta_k}\right)\|g_k\|^2$$

为了获得该算法的全局收敛性，可做如下假设：

假设 3.1 (i)水平集$\Omega = \left\{x \in \Re^n \,\middle|\, f(x) \le f(x_0)\right\}$是有界的，其中$x_0$是初始点。

（ii）在包含于Ω的开凸集Ω_0中，函数$f(x)$是有下界的、可微的且梯度$g(x)$是利普希茨连续的，即存在一个正的常数L，对$\forall\ x, y \in \Re^n$，都有：

$$\|g(x) - g(y)\| \le L\|x - y\| \tag{3.2}$$

根据充分下降性和假设 3.1，与文献[5]中的引理 1.1 相似，可以得到下面的引理，这里省略其证明过程。

引理 3.2 假设 3.1 成立，$\{x_k\}$由 Algorithm N 产生，则对每一个正整数k，都有$f_k \le C_k \le A_k$，并且存在满足非单调线搜索技术的步长α_k。

引理 3.3 假设 3.1 成立，$\{g_k\}$，$\{d_k\}$由 Algorithm N 产生，则有：

$$\alpha_k \ge \frac{1-\sigma}{L}\frac{\left|g_k^T d_k\right|}{\|d_k\|^2}, \tag{3.3}$$

$$\left\|y_k^m\right\| \le 2L\|s_k\|, \tag{3.4}$$

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty \tag{3.5}$$

证明：由式（1.6）、（1.7）、（1.8）及假设 3.1 中的利普希茨连续，有：

$$\alpha_k L\|d_k\|^2 \ge g_{k+1}^T d_k - g_k^T d_k \ge (\sigma - 1) g_k^T d_k ,$$

即

$$\alpha_k \ge \frac{(\sigma-1) g_k^T d_k}{L\|d_k\|^2} = \frac{1-\sigma}{L}\frac{\left|g_k^T d_k\right|}{\|d_k\|^2}$$

由中值定理得：

$$\rho_k = 2\left(f_k - f_{k+1}\right) + \left(g_k + g_{k+1}\right)^T s_k =$$
$$-2g\left(x_k + \theta s_k\right)s_k + \left(g_k + g_{k+1}\right)^T s_k \le$$
$$\left[\left\|g\left(x_k + \theta s_k\right) - g_k\right\| + \left\|g\left(x_k + \theta s_k\right) - g_{k+1}\right\|\right]\|s_k\| \le$$
$$\left[L(1-\theta)\|s_k\| + L\theta\|s_k\|\right]\|s_k\| = L\|s_k\|^2$$

其中$\theta \in (0,1)$。又由y_k^m的定义，可得：

$$\left\|y_k^m\right\| \le \|y_k\| + \frac{|\rho_k|\|s_k\|}{\|s_k\|^2} \le L\|s_k\| + L\|s_k\| = 2L\|s_k\|$$

由式（1.8）可得：

$$Q_{k+1} = \eta_k Q_k + 1 = 1 + \sum_{i=1}^{k}\prod_{j=0}^{i-1}\eta_{k-j} \le$$
$$1 + \sum_{i=1}^{k}\eta_{\max}^i \le \sum_{i=1}^{\infty}\eta_{\max}^i = \frac{1}{1-\eta_{\max}} ,$$

$$C_{k+1} = \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \le \frac{\eta_k Q_k C_k + C_k + \frac{\sigma-1}{L}\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2}}{Q_{k+1}} =$$
$$C_k - \frac{\frac{1-\sigma}{L}\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2}}{Q_{k+1}}$$

即 $\dfrac{\left(g_k^T d_k\right)^2}{\left\|d_k\right\|^2} \le \dfrac{Q_{k+1}L}{1-\sigma}\left(C_k - C_{k+1}\right) \le$

$$\frac{L}{(1-\sigma)(1-\eta_{\max})}\left(C_k - C_{k+1}\right)$$

由函数 f 的有界性及 C_k 的定义和引理 3.2，得出 C_k 有下界，从而 Zoutendijk 条件 $\sum_{k=0}^{\infty}\dfrac{\left(g_k^T d_k\right)^2}{\left\|d_k\right\|^2} < \infty$ 成立。

基于引理 3.1-3.3 及假设 3.1，类似于文献[8]的定理 3.2，得到 Algorithm N 的全局收敛性，这里省略其证明。

定理 3.1 假设 3.1 成立，$\{\alpha_k, d_k, g_k\}$ 由 Algorithm N 产生，则存在正整数 k，使得 $g_k = 0$ 或者 $\lim\limits_{k\to\infty}\inf\left\|g_k\right\| = 0$。

4 数值试验

本节将采用一些 Benchmark problems 来测试 Algorithm N、CG_DESCENT 算法和 HS 算法的数值表现。程序在 MATLAB 2010b 上编写，在处理器为 Dual-Core E5800、3.2GHZ，SDRAM 为 2.0G bytes 的电脑上运行。测试函数列举如下：

函数 1 Sphere function:

$$f(x) = \sum_{i=1}^{p} x_i^2, \qquad x_i \in [-5.12, 5.12]$$

其中最优解 $x^* = (0,0,\cdots,0)$，最优函数值 $f(x^*) = 0$。

函数 2 Schwefel's function :

$$f(x) = \sum_{i=1}^{p}\left(\sum_{j=1}^{i} x_j\right)^2, \qquad x_i \in [-65.536, 65.536]$$

其中最优解 $x^* = (0,0,\cdots,0)$，最优函数值 $f(x^*) = 0$。

函数 3 Rastrigin function:

$$f(x) = 10p + \sum_{i=1}^{p}\left(x_i^2 - 10\cos(2\pi x_i)\right), \ x_i \in [-5.12, 5.12]$$

其中最优解 $x^* = (0,0,\cdots,0)$，最优函数值 $f(x^*) = 0$。

函数 4 Schwefel function:

$$f(x) = 418.9829p + \sum_{i=1}^{p} x_i \sin\left(\sqrt{|x_i|}\right),$$

$$x_i \in [-512.03, 511.97]$$

其中最优解 $x^* = (-420.9687, -420.9687, \cdots, -420.9687)$，最优函数值 $f(x^*) = 0$。

函数 5 Grewank function:

$$f(x) = 1 + \sum_{i=1}^{p}\frac{x_i^2}{400} - \prod_{i=1}^{p}\cos\left(\frac{x_i}{\sqrt{i}}\right), \quad x_i \in [-600, 600]$$

其中最优解 $x^* = (0,0,\cdots,0)$，最优函数值 $f(x^*) = 0$。

在测试过程中，使用 Himmeblau 终止准则：即如果 $|f(x_k)| \le \varepsilon_2$，令 $stop = |f(x_k) - f(x_{k+1})|$；否则，令 $stop = \dfrac{|f(x_k) - f(x_{k+1})|}{|f(x_k)|}$。如果 $\|g(x)\| < \varepsilon_1$ 或者 $stop < \varepsilon_3$ 成立，我们将终止程序；与此同时如果总的迭代次数超过 1000 时，我们也将终止程序。试验过程中参数设置如下：

$\delta = 0.1,\ \sigma = 0.9,\ \omega = 0.001,\ \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon = 10^{-6},$

$\eta_k = 0.7,\ \theta_k = 1.$

试验结果如表 1-表 3，其中 Dim 表示问题的维数；x_0 表示初始迭代值；NI 表示迭代的次数；NF 表示函数值计算次数；f 表示迭代停止时函数值。

表 1 Algorithm N 的数值结果

Table 1 Test results for Algorithm N

Sphere	x_0	NI/NF/f (-2,-2,...,-2)	NI/NF/f (2,2,...,2)	NI/NF/f (-2,0,-2,0,...)	NI/NF/f (2,0,2,0,...)
	30	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000
Dim	500	2/2/4.831773e-027	2/2/4.831773e-027	2/2/2.415887e-027	2/2/2.415887e-027
	1000	2/2/2.415887e-027	2/2/2.415887e-027	2/2/1.207943e-027	2/2/1.207943e-027
Schwefel's	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	8/8/3.841514e-007	8/8/3.841514e-007	12/12/6.366277e-007	12/12/6.366277e-007
Dim	100	16/16/9.387227e-007	16/16/9.387227e-07	33/33/5.118726e-007	33/33/5.118726e-007
	300	7/7/1.063914e-001	7/7/1.063914e-001	244/244/8.875045e-007	244/244/8.875045e-007
Rastrigin	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000
Dim	500	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000
	1000	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000	2/2/0.000000e+000
schwefel	x_0	(-420.5,-420.5,...)	(-420.1,-420.1,..)	(-420.5,0,-420.5,0,...)	(-420.1,0,-420.1,0..)
	30	2/2/3.818270e-004	2/2/3.818270e-004	1/1/6.285159e+003	1/1/6.286172e+003
Dim	500	2/2/6.363784e-003	3/3/6.363784e-003	1/1/1.047527e+005	1/1/1.047695e+005
	1000	2/2/1.272757e-002	3/3/1.272757e-002	1/1/2.095053e+005	1/1/2.095391e+005
Grewank	x_0	(-21,-21,...,-21)	(32,32,...,32)	(-21,0,-21,0,...)	(32,0,32,0,...)
	30	7/53/1.283343e+000	13/75/0.00000e+000	36/251/1.742645e+000	5/35/8.729174e-001
Dim	500	3/18/3.163330e-010	3/3/2.553291e-012	3/3/1.865093e-010	3/3/2.484568e-012
	1000	3/18/3.710632e-011	3/3/5.963452e-012	3/18/6.275258e-011	3/3/5.206946e-013

表 2 CG_DESCENT 算法数值结果

Table 2 Test results for CG_DESCENT method

Sphere	x_0	NI/NF/f (-2,-2,...,-2)	NI/NF/f (2,2,...,2)	NI/NF/f (-2,0,-2,0,...)	NI/NF/f (2,0,2,0,...)
	30	2/2/8.590273e-020	2/2/8.590273e-020	2/2/4.295136e-020	2/2/4.295136e-020
Dim	500	2/2/1.431712e-018	2/2/1.431712e-018	2/2/7.158561e-019	2/2/7.158561e-019
	1000	2/2/2.863424e-01	2/2/2.863424e-018	2/2/1.431712e-018	2/2/1.431712e-018
Schwefel's	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	23/23/9.213356e-007	23/23/9.213356e-007	16/31/2.806430e-006	16/31/2.806430e-006
Dim	100	36/51/1.595691e-005	36/51/1.595691e-005	34/49/9.606893e-006	34/49/9.606893e-006
	300	17/227/NaN	17/227/NaN	40/55/1.489226e-004	40/55/1.489226e-004
Rastrigin	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	3/3/2.811129e-008	3/3/2.811129e-008	3/3/7.027950e-009	3/3/7.027950e-009
Dim	500	4/4/0.000000e+000	4/4/0.000000e+000	4/4/0.000000e+000	4/4/0.000000e+000
	1000	4/4/0.000000e+000	4/4/0.000000e+000	4/4/0.000000e+000	4/4/0.000000e+000
schwefel	x_0	(-420.5,-420.5,...)	(-420.1,-420.1,..)	(-420.5,0,-420.5,0,...)	(-420.1,0,-420.1,0..)
	30	4/4/3.818270e-004	5/5/3.818270e-004	1/1/6.286172e+003	1/1/6.286172e+003
Dim	500	5/5/6.363784e-003	5/5/6.363784e-003	1/1/1.047527e+005	1/1/1.047695e+005
	1000	5/5/1.272757e-002	5/5/1.272757e-002	1/1/2.095053e+005	1/1/2.095391e+005
Grewank	x_0	(-21,-21,...,-21)	(32,32,...,32)	(-21,0,-21,0,...)	(32,0,32,0,...)
	30	63/396/1.767733e+000	28/118/6.363065e-001	41/297/2.798860e+000	7/22/1.342944e+000
Dim	500	2/2/4.286571e-013	2/2/4.996004e-014	2/2/2.353673e-013	2/2/2.819966e-014
	1000	2/2/4.994893e-013	2/2/5.195844e-014	2/2/2.708944e-013	2/2/2.908784e-014

表 3 HS 算法的数值结果

Table 3 Test results for HS method

Sphere	x_0	NI/NF/f(-2,-2,...,-2)	NI/NF/f(2,2,...,2)	NI/NF/f(-2,0,-2,0,...)	NI/NF/f(2,0,2,0,...)
	30	2/2/8.590273e-020	2/2/8.590273e-020	2/2/4.295136e-020	2/2/4.295136e-020
Dim	500	2/2/1.431712e-018	2/2/1.431712e-018	2/2/7.158561e-019	2/2/7.158561e-019
	1000	2/2/2.863424e-018	2/2/2.863424e-018	2/2/1.431712e-018	2/2/1.431712e-018
Schwefel's	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	6/21/7.409251e-005	6/21/7.409251e-005	6/21/1.852107e-005	6/21/1.852107e-005
Dim	100	6/21/2.664951e-003	6/21/2.664951e-003	6/21/6.537876e-004	6/21/6.537876e-004
	300	6/21/7.127432e-002	6/21/7.127432e-002	6/21/1.767486e-002	6/21/1.767486e-002
Rastrigin	x_0	(-0.001,-0.001,...)	(0.001,0.001,...)	(-0.001,0,-0.001,0,...)	(0.001,0,0.001,0,...)
	30	3/18/1.023182e-012	3/18/1.023182e-012	3/3/5.115908e-013	3/3/5.115908e-013
Dim	500	3/3/2.728484e-012	3/3/2.728484e-012	3/3/1.818989e-012	3/3/1.818989e-012
	1000	3/18/3.637979e-012	3/18/3.637979e-012	3/3/1.818989e-012	3/3/1.818989e-012
schwefel	x_0	(-420.5,-420.5,...)	(-420.1,-420.1,..)	(-420.5,0,-420.5,0,...)	(-420.1,0,-420.1,0..)
	30	3/18/3.819093e-004	3/3/3.830492e-004	1/1/6.285159e+003	1/1/6.286172e+003
Dim	500	3/3/6.365155e-003	3/18/6.384153e-003	1/1/1.047527e+005	1/1/1.047695e+005
	1000	3/3/1.273031e-002	3/18/1.276831e-002	1/1/2.095053e+005	1/1/2.095391e+005
Grewank	x_0	(-21,-21,...,-21)	(32,32,...,32)	(-21,0,-21,0,...)	(32,0,32,0,...)
	30	4/34/1.432366e+000	10/55/9.514236e+001	8/98/6.723494e+001	3/18/3.918663e+000
Dim	500	2/2/4.286571e-013	2/2/4.996004e-014	2/2/2.353673e-013	2/2/2.819966e-014
	1000	2/2/4.994893e-013	2/2/5.195844e-014	2/2/2.708944e-013	2/2/2.908784e-014

对于上述 5 个问题的测试，由表 1-表 3 可以看出：Algorithm N 的数值表现要优于 CG_DESCENT 算法的数值表现，同时也优于 HS 算法的数值表现，由此可看出 Algorithm N 比 CG_DESCENT 算法和 HS 算法更有效。由此也说明了 Algorithm N 对于求解无约束优化问题的有效性。

参考文献：

[1] Fletcher R, Reeves C M. Function minimization by conjugate gradients[J]. The computer journal,1964, 7(2): 149-154.

[2] Hestenes M R, Stiefel E. Methods of conjugate gradients for solving linear systems[J]. NBS,1952,49:409-436.

[3] Wei Z, Yao S, Liu L. The convergence properties of some new conjugate gradient methods[J]. Applied Mathematics and Computation, 2006, 183(2): 1341-1350.

[4] Shengwei Y, Wei Z, Huang H. A note about WYL's conjugate gradient method and its applications[J]. Applied Mathematics and computation, 2007, 191(2): 381-388.

[5] Zhang H, Hager W W. A nonmonotone line search technique and its application to unconstrained optimization[J]. SIAM Journal on Optimization, 2004, 14(4): 1043-1056.

[6] Gilbert J C, Nocedal J. Global convergence properties of conjugate gradient methods for optimization[J]. SIAM Journal on optimization, 1992, 2(1): 21-42.

[7] Hu Y F, Storey C. Global convergence result for conjugate gradient methods[J]. Journal of Optimization Theory and Applications, 1991, 71(2): 399-405.

[8] Hager W W, Zhang H. A new conjugate gradient method with guaranteed descent and an efficient line search[J]. SIAM Journal on Optimization, 2005, 16(1): 170-192.

[9] Hager W W, Zhang H. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent[J]. ACM Transactions on Mathematical Software (TOMS), 2006, 32(1): 113-137.

[10] Yuan G, Lu X, Wei Z. A conjugate gradient method with descent direction for unconstrained optimization[J]. Journal of Computational and Applied Mathematics, 2009, 233(2): 519-530.

[11] Yuan G, Wei Z, Li G. A modified Polak–Ribière–Polyak conjugate gradient algorithm for nonsmooth convex programs[J]. Journal of Computational and Applied Mathematics, 2014, 255: 86-96.

[12] Hager W W, Zhang H. A survey of nonlinear conjugate gradient methods[J]. Pacific journal of Optimization, 2006, 2(1): 35-58.

[13] Wei Z, Yu G, Yuan G, et al. The superlinear convergence of a modified BFGS-type method for unconstrained optimization[J]. Computational optimization and applications, 2004, 29(3): 315-332.

[14] Yuan G , Wei Z. Convergence analysis of a modified BFGS method on convex minimizations [J]. Computational Optimization and Applications, 2010, 47(2): 237-255.

[15] Andrei N. Acceleration of conjugate gradient algorithms for unconstrained optimization[J]. Applied Mathematics and Computation, 2009, 213(2): 361-369.

[16] Andrei N. Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization[J].Journal of Optimization Theory and Applications,2013,159(1): 159-182.