COMPUTATIONAL & APPLIED MATHEMATICS

Volume 24, N. 3, pp. 399–416, 2005

ISSN 0101-8205
www.scielo.br/cam

Step-size Estimation for Unconstrained Optimization Methods

ZHEN-JUN SHI[1,2] and JIE SHEN[3]

[1]College of Operations Research and Management, Qufu Normal University,
Rizhao, Shandong 276826, P.R.China
[2]Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
P.O. Box 2719, Beijing 100080, China
[3]Department of Computer & Information Science,
University of Michigan, Dearborn, MI 48128, USA

E-mails: zjshi@qrnu.edu.cn / zjshi@lsec.cc.ac.cn / shen@umich.edu

Abstract. Some computable schemes for descent methods without line search are proposed. Convergence properties are presented. Numerical experiments concerning large scale unconstrained minimization problems are reported.

Mathematical subject classification: 90C30, 65K05, 49M37.

Key words: unconstrained optimization, descent methods, step-size estimation, convergence.

1 Introduction

A well-known algorithm, for the unconstrained minimization of a function $f(x)$ in n variables

$$f: R^n \to R, \tag{1}$$

having Lipschitz continuous first partial derivatives, is the steepest descent method (Fiacco, 1990, Polak, 1997, [8, 17]). The iterations correspond to the following equation

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \ k = 0, 1, 2, ..., \tag{2}$$

#620/04. Received: 29/XI/04. Accepted: 16/V/05.

where α_k is the smallest nonnegative value of α that locally minimizes f along the direction $-\nabla f(x_k)$ starting from x_k. Curry (Curry, 1944, [5]) showed that any limit point x^* of the sequence $\{x_k\}$ generated by (2) is a stationary point ($\nabla f(x^*) = 0$).

The iterative scheme (2) is not practical because the step-size rule at each step involves an exact one-dimensional minimization problem. However, the steepest descent algorithm can be implemented by using inexact one-dimensional minimization. The first efficient inexact step-size rule was proposed by Armijo (Armijo, 1966, [1]). It can be shown that, under mild assumptions and with different step-size rules, the iterative scheme (2) converges to a local minimizer x^* or a saddle point of $f(x)$, but its convergence is only linear and sometimes slower than linear.

The steepest descent method is particularly useful when the dimension of the problem is very large. However, it may generate short zigzagging displacements in a neighborhood of a solution (Fiacco, 1990, [8]).

For simplicity, we denote $\nabla f(x_k)$ by g_k, $f(x_k)$ by f_k and $f(x^*)$ by f^*, respectively, where x^* denotes a local minimizer of f. In the algorithmic framework of steepest descent methods, Goldstein (Goldstein, 1962, 1965, 1967, [10, 11, 12]) investigated the iterative formula

$$x_{k+1} = x_k + \alpha_k d_k, \ k = 0, 1, 2, ..., \tag{3}$$

where d_k satisfies the relation

$$g_k^T d_k < 0, \tag{4}$$

which guarantees that d_k is a descent direction of $f(x)$ at x_k (Cohen, 1981, Nocedal and Wright, 1999, [4], [14]). In order to guarantee the global convergence, it is usually required to satisfy the descent condition

$$g_k^T d_k \leq -c\|g_k\|^2, \tag{5}$$

where $c > 0$ is a constant. The angle property

$$\cos\langle -g_k, d_k\rangle = -\frac{g_k^T d_k}{\|g_k\| \cdot \|d_k\|} \geq \eta_0, \tag{6}$$

is often used in many situations, with $\eta_0 \in (0, 1]$.

Observe that, if $\|g_k\| \neq 0$ then $d_k = -g_k$ satisfies (4), (5) and (6) simultaneously. Throughout this paper, we take $d_k = -g_k$.

There are many alternative line-search rules to choose α_k along the ray $S_k = \{x_k + \alpha d_k | \ \alpha > 0\}$. Namely:

(a) Minimization Rule. At each iteration, α_k is selected so that

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k). \tag{7}$$

(b) Approximate Minimization Rule. At each iteration, α_k is selected so that

$$\alpha_k = \min\{\alpha | g(x_k + \alpha d_k)^T d_k = 0, \alpha > 0\}. \tag{8}$$

(c) Armijo Rule. Set scalars s_k, γ, L and σ with $s_k = -\frac{g_k^T d_k}{L\|d_k\|^2}$, $\gamma \in (0, 1)$, $L > 0$ and $\sigma \in (0, \frac{1}{2})$. Let α_k be the largest α in $\{s_k, \gamma s_k, \gamma^2 s_k, ..., \}$ such that

$$f_k - f(x_k + \alpha d_k) \geq -\sigma \alpha g_k^T d_k. \tag{9}$$

(d) Limited Minimization Rule. Set $s_k = -\frac{g_k^T d_k}{L\|d_k\|^2}$ where α_k is defined by

$$f(x_k + \alpha_k d_k) = \min_{\alpha \in [0, s_k]} f(x_k + \alpha d_k), \tag{10}$$

and $L > 0$ is a given constant.

(e) Goldstein Rule. A fixed scalar $\sigma \in (0, \frac{1}{2})$ is selected, and α_k is chosen in order to satisfy

$$\sigma \leq \frac{f(x_k + \alpha_k d_k) - f_k}{\alpha_k g_k^T d_k} \leq 1 - \sigma. \tag{11}$$

(f) Strong Wolfe Rule. At the k-th iteration, α_k satisfies simultaneously

$$f_k - f(x_k + \alpha_k d_k) \geq -\sigma \alpha_k g_k^T d_k \tag{12}$$

and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\beta g_k^T d_k, \tag{13}$$

where $\sigma \in (0, \frac{1}{2})$ and $\beta \in (\sigma, 1)$.

(g) Wolfe Rule. At the k-th iteration, α_k satisfies (12) and

$$g(x_k + \alpha_k d_k)^T d_k \geq \beta g_k^T d_k. \tag{14}$$

Some important global convergence results for methods using the above mentioned specific line search procedures have been given in the literature ([25, 15, 16, 23, 24]).

This paper is organized as follows. In the next section we describe some descent algorithms without line search. In Sections 3 and 4 we analyze their global convergence and convergence rate respectively. In Section 5 we give some numerical experiments and conclusions.

2 Descent Algorithm without Line Search

We assume that

(H1). $f(x)$ is bounded below. We denote $L(x_0) = \{x \in R^n | f(x) \leq f(x_0)\}$.

(H2). The gradient $g(x)$ is uniformly continuous on an open convex set B that contains L_0.

We sometimes further assume that the following condition holds.

(H3). The gradient $g(x)$ is Lipschitz continuous on an open convex set B that contains the level set $L(x_0)$, i.e., there exists L such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \ \forall x, y \in B. \tag{15}$$

Obviously, $(H3)$ implies $(H2)$.

We shall implicitly assume that the constant L in (H3) is easy to estimate.

Algorithm (A).

Step 0. Choose $x_0 \in R^n$, $\delta \in (0, 2)$ and $L_0 > 0$ and set $k := 0$;

Step 1. If $\|g_k\| = 0$ then stop; else go to Step 2;

Step 2. Estimate $L_k > 0$;

Step 3. $x_{k+1} = x_k - \frac{\delta}{L_k} g_k$;

Step 4. Set $k := k + 1$ and go to Step 1.

Note. In the above algorithm, line search procedure is avoided at each iteration, which may reduce the cost of computation. However, we must estimate L_k at each iteration. Certainly, if the Lipschitz constant L of the gradient of objective functions is known a priori, then we can take $L_k \equiv L$ in the algorithm. In many practical problems, the Lipschitz constant L is not known a priori and we must estimate it and find an approximation L_k to L at each step.

For estimating L_k we define

$$L_k = \max\left(L_{k-1}, \frac{\|g_k - g_{k-1}\|}{\|x_k - x_{k-1}\|}\right), \ k \geq 1. \tag{16}$$

We can also estimate L_k in Algorithm (A) by solving the following minimization problem

$$\min_{L \in R^1} \|L\delta_{k-1} - y_{k-1}\|, \tag{17}$$

where $\delta_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$ and $\|\cdot\|$ denotes Euclidean norm. In this case,

$$L_k = \max\left(L_{k-1}, \frac{y_{k-1}^T \delta_{k-1}}{\|\delta_{k-1}\|^2}\right), \ k \geq 1. \tag{18}$$

Similarly, L_k can be found by solving the problem

$$\min_{L \in R^1} \left\|y_{k-1} - \frac{1}{L_k}\delta_k\right\| \tag{19}$$

and set

$$L_k = \max\left(L_{k-1}, \frac{\|y_{k-1}\|^2}{y_{k-1}^T \delta_{k-1}}\right), \ k \geq 1. \tag{20}$$

The last two formulae are useful because they arise from the classical quasi-Newton condition (e.g. [14]) and from Barzilai and Borwein's idea (1988, [2]). Some recent observations on Barzilai and Borwein's method are very exciting (Fletcher, 2001, [7], Raydan 1993, 1997, [20, 21], Dai and Liao, 2002, [6]).

If the Hessian matrix $\nabla^2 f(x_k)$ is easy to evaluate then we can take

$$L_k = \max\{L_{k-1}, \|\nabla^2 f(x_k)\|\}, \ k \geq 1. \tag{21}$$

3 Convergence Analysis

The following lemma can be found in many text books. See, for example, [14].

Lemma 3.1 (mean value theorem). *Suppose that the objective function $f(x)$ is continuously differentiable on an open convex set B, then*

$$f(x_k + \alpha d_k) - f_k = \alpha \int_0^1 d_k^T g(x_k + t\alpha d_k) dt, \tag{22}$$

where $x_k, x_k + \alpha d_k \in B$ and $d_k \in R^n$. Further, if $f(x)$ is twice continuously differentiable on B, then

$$g(x_k + \alpha d_k) - g_k = \alpha \int_0^1 \nabla^2 f(x_k + t\alpha d_k) d_k dt \tag{23}$$

and

$$f(x_k + \alpha d_k) - f_k = \alpha g_k^T d_k + \alpha^2 \int_0^1 (1-t) d_k^T \nabla^2 f(x_k + t\alpha d_k) d_k dt. \tag{24}$$

3.1 Convergence of Algorithm (A)

Theorem 3.1. *If $(H1)$ and $(H3)$ hold, Algorithm (A) generates an infinite sequence $\{x_k\}$, and*

$$\rho \in \left(\frac{\delta}{2}, 1\right), \; L_k \geq \rho L, \; \sum_{k=0}^{+\infty} \frac{1}{L_k^2} = +\infty, \tag{25}$$

then

$$\liminf_{k \to +\infty} \|g_k\| = 0. \tag{26}$$

Proof. By Lemma 3.1 and $(H3)$ we have

$$\begin{aligned} f(x_k + \alpha d_k) - f_k &= \alpha \int_0^1 d_k^T g(x_k + t\alpha d_k) dt \\ &= \alpha g_k^T d_k + \alpha \int_0^1 d_k^T (g(x_k + t\alpha d_k) - g_k) dt \\ &\leq \alpha g_k^T d_k + \alpha \int_0^1 \|d_k\| \cdot \|g(x_k + t\alpha d_k) - g_k\| dt \\ &\leq \alpha g_k^T d_k + \alpha^2 L \int_0^1 t \|d_k\|^2 dt \\ &= \alpha g_k^T d_k + \frac{1}{2} \alpha^2 L \|d_k\|^2. \end{aligned}$$

Taking $d_k = -g_k$ and $\alpha = \frac{\delta}{L_k}$ in the above formula, we have

$$\begin{aligned} f(x_k - \alpha g_k) - f_k &\leq -\alpha_k \|g_k\|^2 + \frac{1}{2}\alpha_k^2 L \|g_k\|^2 \\ &= -\frac{\delta \|g_k\|^2}{L_k} + \frac{\delta^2 L}{2L_k^2}\|g_k\|^2 \\ &\leq -(\frac{\delta}{L_k} - \frac{\delta^2 L}{2L_k^2})\|g_k\|^2. \end{aligned}$$

Noting that $L_k \geq \rho L$, we obtain

$$\begin{aligned} \frac{\delta}{L_k} - \frac{\delta^2 L}{2L_k^2} &= \frac{2\delta L_k - \delta^2 L}{2L_k^2} \\ &\geq \frac{(2\rho - \delta)\delta L}{2L_k^2} \\ &> 0. \end{aligned}$$

Therefore, $\{f_k\}$ is a monotone decreasing sequence. So, by $(H1)$, $\{f_k\}$ has a lower bound and, thus, $\{f_k\}$ has a finite limit. It follows from the above inequality that

$$\sum_{k=0}^{\infty} \frac{(2\rho - \delta)L}{2L_k^2}\|g_k\|^2 < +\infty. \tag{27}$$

By (25) we have

$$\sum_{k=0}^{\infty} \frac{(2\rho - \delta)L}{2L_k^2} = +\infty.$$

The above inequality and (27) show that (26) holds. The proof is finished. □

Corollary 3.1. *If the conditions in Theorem 3.1 hold and $\rho L \leq L_k \leq M$ ($M > 0$ is a fixed large integer) for all k, then*

$$\sum_{k=0}^{\infty} \|g_k\|^2 < +\infty, \tag{28}$$

and, thus,

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{29}$$

Remark. The above theorem shows that we can set a large L_k to guarantee the global convergence. However, if L_k is very large then α_k will be very small and will slow the convergence rate of descent methods. On the other hand, very small values of L_k may fail to guarantee the global convergence. Thus, it is better to set an adequate estimation L_k at each iteration.

3.2 Comparing with other step-sizes

Theorem 3.2. *Assume that the hypotheses of Theorem 3.1 hold. Denote the exact step-size by α_k^* (including exact line search rules (a) and (b)). Then*

$$\alpha_k^* \geq \frac{\rho}{\delta}\alpha_k. \tag{30}$$

Proof. For the line search rules (a) and (b), $(H3)$ and Cauchy-Schwartz inequality, we have:

$$\begin{aligned} \alpha_k^* L \|g_k\|^2 &\geq \|g_{k+1} - g_k\| \cdot \|g_k\| \\ &\geq -(g_{k+1} - g_k)^T g_k \\ &= \|g_k\|^2. \end{aligned}$$

Therefore,

$$\alpha_k^* \geq \frac{1}{L}.$$

Noting that $L_k \geq \rho L$, we have

$$\alpha_k^* \geq \frac{1}{L} \geq \frac{\rho}{L_k} = \frac{\delta}{L_k} \cdot \frac{\rho}{\delta} = \alpha_k \cdot \frac{\rho}{\delta}. \qquad \square$$

Theorem 3.3. *Assume that the hypotheses of Theorem 3.1 hold. Denote α_k^* the step-size defined by the line search rule (c) with L being the Lipschitz constant of $\nabla f(x)$. Then,*

$$\alpha_k^* \geq \frac{\rho}{\delta}\alpha_k,\ k \in K_1;\ \ \alpha_k^* \geq \frac{\rho\gamma(1-\sigma)}{\delta}\alpha_k,\ k \in K_2, \tag{31}$$

where $K_1 = \{k|\ \alpha_k^* = s_k\}$ and $K_2 = \{k|\ \alpha_k^* < s_k\}$.

Proof. If $k \in K_1$, then

$$\alpha_k^* = s_k = \frac{1}{L} = \frac{\rho}{\rho L} \geq \frac{\rho}{\delta} \cdot \frac{\delta}{L_k} = \frac{\rho}{\delta}\alpha_k.$$

If $k \in K_2$ then $\alpha_k^* < s_k$ and thus $\alpha_k^*/\gamma \leq s_k$, by line search rule (c), we have

$$f(x_k + \alpha_k^* d_k/\gamma) - f_k > \sigma(\alpha_k^*/\gamma) g_k^T d_k.$$

Using the Mean Value Theorem on the left-hand side of the above inequality, we see that there exists $\theta_k \in [0, 1]$ such that

$$(\alpha_k^*/\gamma) g(x_k + \theta_k \alpha_k^* d_k/\gamma)^T d_k > \sigma(\alpha_k^*/\gamma) g_k^T d_k,$$

and, thus,

$$g(x_k + \theta_k \alpha_k^* d_k/\gamma)^T d_k > \sigma g_k^T d_k. \tag{32}$$

By $(H3)$, Cauchy-Schwartz inequality and (32) we obtain:

$$\begin{aligned} \alpha_k L \|d_k\|/\gamma &\geq \|g(x_k + \theta_k \alpha_k^* d_k/\gamma) - g_k\| \cdot \|d_k\| \\ &\geq [g(x_k + \theta_k \alpha_k^* d_k/\gamma) - g_k]^T d_k \\ &\geq -(1-\sigma) g_k^T d_k. \end{aligned}$$

Since $d_k = -g_k$, by the above inequality, we get:

$$\alpha_k^* \geq \frac{\gamma(1-\sigma)}{L} \geq \frac{\rho\gamma(1-\sigma)}{\delta}\alpha_k. \qquad \square$$

Theorem 3.4. *Assume that the hypotheses of Theorem 3.1 hold. Denote α_k^* the step-size defined by the line search rule (d) with L being the Lipschitz constant of $\nabla f(x)$. Then,*

$$\alpha_k^* \geq \frac{\rho}{\delta}\alpha_k, \tag{33}$$

where α_k is yet the step-size of Algorithm (A).

Proof. If $\alpha_k^* = s_k$ then

$$\alpha_k^* = \frac{1}{L} = \frac{\rho}{\rho L} \geq \frac{\rho}{L_k} = \frac{\rho}{\delta}\alpha_k.$$

If $\alpha_k^* < s_k$ then $g_{k+1}^T d_k = 0$ and, thus,

$$\alpha_k^* L \|d_k\|^2 \geq \|g_{k+1} - g_k\| \cdot \|d_k\| \geq -g_k^T d_k.$$

Since $d_k = -g_k$, by the above inequality, we get:

$$\alpha_k^* \geq \frac{1}{L} \geq \frac{\rho}{\rho L} \geq \frac{\rho}{L_k} = \frac{\rho}{\delta}\alpha_k. \qquad \square$$

Theorem 3.5. *Assume that the hypotheses of Theorem 3.1 hold and let α_k^* be defined by the line search rule (e). Then,*

$$\alpha_k^* \geq \frac{\rho\sigma}{\delta} \cdot \alpha_k, \tag{34}$$

Proof. By the line search rule (e), we have

$$f(x_k + \alpha_k^* d_k) - f_k \geq (1-\sigma)\alpha_k^* g_k^T d_k.$$

By the mean value theorem, there exists θ_k such that

$$\alpha_k^* g(x_k + \theta_k \alpha_k^* d_k)^T d_k \geq (1-\sigma)\alpha_k^* g_k^T d_k.$$

So,

$$g(x_k + \theta_k \alpha_k^* d_k)^T d_k \geq (1-\sigma) g_k^T d_k. \tag{35}$$

By $(H3)$, the Cauchy-Schwartz inequality and (36), we have:

$$\begin{aligned} \alpha_k^* L \|d_k\|^2 &\geq \|g(x_k + \theta_k \alpha_k^* d_k) - g_k\| \cdot \|d_k\| \\ &\geq [g(x_k + \theta_k \alpha_k^* d_k) - g_k]^T d_k \\ &\geq -\sigma g_k^T d_k. \end{aligned}$$

Since $d_k = -g_k$, we get:

$$\alpha_k^* \geq \frac{\sigma}{L} = \frac{\rho\sigma}{\rho L} \geq \frac{\rho\sigma}{L_k} = \frac{\rho\sigma}{\delta}\alpha_k. \qquad \square$$

Theorem 3.6. *Assume that the hypotheses of Theorem 3.1 hold and that α_k^* is defined by the line search rules (f) or (g). Then,*

$$\alpha_k^* \geq \frac{\rho(1-\beta)}{\delta} \cdot \alpha_k, \tag{36}$$

Proof. By (13), (14) and the Cauchy-Schwartz inequality, we have:

$$\begin{aligned} \alpha_k^* L\|d_k\|^2 &\geq \|g_{k+1} - g_k\| \cdot \|d_k\| \\ &\geq [g_{k+1} - g_k]^T d_k \\ &\geq -(1-\beta) d_k^T d_k. \end{aligned}$$

Since $d_k = -g_k$ we get:

$$\alpha_k^* \geq \frac{1-\beta}{L} = \frac{\rho(1-\beta)}{\rho L} \geq \frac{\rho(1-\beta)}{L_k} = \frac{\rho(1-\beta))}{\delta}\alpha_k. \qquad \square$$

4 Convergence Rate

In order to analyze the convergence rate of the algorithm, we make use of Assumption (H_4) below.

(H_4). $\{x_k\} \to x^*(k \to \infty)$, $f(x)$ is twice continuously differentiable on $N(x^*, \epsilon)$ and $\nabla^2 f(x^*)$ is positive definite.

Lemma 4.1. *Assume that (H_4) holds. Then, $(H1)$, $(H3)$ and thus $(H2)$ hold automatically for k sufficiently large, and there exists $0 < m' \leq M'$ and $\epsilon_0 \leq \epsilon$ such that*

$$m'\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M'\|y\|^2, \ \forall x, y \in N(x^*, \epsilon_0); \tag{37}$$

$$\frac{1}{2}m'\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2}M'\|x - x^*\|^2, \ \forall x \in N(x^*, \epsilon_0); \tag{38}$$

$$M'\|x-y\|^2 \geq (g(x)-g(y))^T(x-y) \geq m'\|x-y\|^2, \ \forall x, y \in N(x^*, \epsilon_0); \tag{39}$$

and thus

$$M'\|x - x^*\|^2 \geq g(x)^T(x - x^*) \geq m'\|x - x^*\|^2, \ \forall x \in N(x^*, \epsilon_0). \tag{40}$$

By (39) and (40) we can also obtain, from Cauchy-Schwartz inequality, that

$$M'\|x - x^*\| \geq \|g(x)\| \geq m'\|x - x^*\|, \ \forall x \in N(x^*, \epsilon_0), \tag{41}$$

and

$$\|g(x) - g(y)\| \leq M'\|x - y\|, \forall x, y \in N(x^*, \epsilon_0). \tag{42}$$

The proof of this theorem results from Lemma 2.2.7 of [25]. See also Lemma 3.1.4 of [26].

Lemma 4.2. *If* $(H1)$ *and* $(H3)$ *hold and Algorithm (A) with* $L_k \leq M$ *and* $L < M$ *generates an infinite sequence* $\{x_k\}$*, then there exists* $\eta > 0$ *such that*

$$f_k - f_{k+1} \geq \eta \|g_k\|^2, \ \forall k. \tag{43}$$

Proof. As in the proof of Theorem 3.1, we have:

$$\begin{aligned} f_k - f_{k+1} &\geq (\frac{\delta}{L_k} - \frac{\delta^2 L}{2L_k^2})\|g_k\|^2 \\ &\geq \frac{(2\rho - \delta)\delta L}{2M^2}\|g_k\|^2. \end{aligned}$$

Taking

$$\eta = \frac{(2\rho - \delta)\delta L}{2M^2},$$

we obtain the desired result. □

Theorem 4.1. *If the assumption* (H_4) *holds, Algorithm (A) with* $\rho L \leq L_k \leq M$ *and* $L \leq M$ *generates an infinite sequence* $\{x_k\}$*, then* $\{x_k\}$ *converges to* x^* *at least R-linearly.*

Proof. By (H4), there exists k' such that $x_k \in N(x^*, \epsilon_0)$ for $k \geq k'$. By (43) and Lemma 4.1 we obtain

$$\begin{aligned} f_k - f_{k+1} &\geq \eta \|g_k\|^2 \\ &\geq \eta m'^2 \|x_k - x^*\|^2 \\ &\geq \frac{2\eta m'^2}{M'}(f_k - f^*), \ k \geq k'. \end{aligned}$$

By (42) we can assume that $M' \leq L$ and prove that $\theta < 1$. In fact, by the definition of η in the proof of Lemma 4.2, we obtain

$$\begin{aligned} \theta^2 &= \frac{2m'^2\eta}{M'} \leq \frac{2m'^2(2\rho - \delta)\delta L}{2M^2 M'} \leq \frac{m'^2(2\sigma - \delta)\delta}{M'^2} \leq (2\rho - \delta)\delta\frac{m'^2}{M'^2} \\ &\leq (2\rho - \delta)\delta = 2\rho\delta - \delta^2 = -\delta^2 - \rho^2 + 2\rho\delta + \rho^2 = \rho^2 - (\rho - \delta)^2 \\ &\leq \rho^2 < 1. \end{aligned}$$

Set

$$\omega = \sqrt{1-\theta^2}.$$

Since, obviously, $\omega < 1$, we obtain from the above inequality that

$$\begin{aligned} f_{k+1} - f^* &\leq (1-\theta^2)(f_k - f^*) \\ &= \omega^2(f_k - f^*) \\ &\leq \ldots \\ &\leq \omega^{2(k-k')}(f_{k'+1} - f^*). \end{aligned}$$

By Lemma 4.1 and the above inequality we have:

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \frac{2}{m'}(f_{k+1} - f^*) \\ &\leq \omega^{2(k-k')}\frac{2(f_{k'+1} - f^*)}{m'}. \end{aligned}$$

Thus,

$$\|x_k - x^*\| \leq \omega^k \sqrt{\frac{2(f_{k'+1} - f^*)}{m'\omega^{2(k'+1)}}}.$$

This shows that $\{x_k\}$ converges to x^* at least R-linearly. □

5 Numerical Experiments

We give an implementable version of this descent method.

Algorithm (A)′.

Step 0. Choose $x_0 \in R^n$, $\delta \in (0, 2)$ and $M \gg L_0 > 0$ and set $k := 0$;

Step 1. If $\|g_k\| = 0$ then stop; else go to Step 2;

Step 2. Estimate $L_k \in [L_0, M]$;

Step 3. $x_{k+1} = x_k - \frac{\delta}{L_k} g_k$;

Step 4. Set $k := k + 1$ and go to Step 1.

The following formulae for $\alpha_k = \delta / L_k (k \geq 1)$:

1.

$$L_k = \min\left(M, \max\left\{L_{k-1}, \frac{y_{k-1}^T \delta_{k-1}}{\|\delta_{k-1}\|^2}\right\}\right) \tag{44}$$

2.
$$L_k = \min\left(M, \max\left\{L_{k-1}, \frac{\|y_{k-1}\|^2}{y_{k-1}^T \delta_{k-1}}\right\}\right) \tag{45}$$

3.
$$L_k = \min\left(M, \max\left\{L_{k-1}, \frac{\|y_{k-1}\|}{\|\delta_{k-1}\|}\right\}\right) \tag{46}$$

4.
$$L_k = \min\left(M, \frac{2(f_k - f_{k-1} + \alpha_{k-1}\|g_{k-1}\|^2)}{\alpha_{k-1}^2 \|g_{k-1}\|^2}\right) \tag{47}$$

are tried to compare the new algorithm against PR conjugate gradient method with restart and BB method ([2]). In conjugate gradient methods one has:

$$d_k = \begin{cases} -g_k, & \text{if } k = 0; \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{48}$$

where

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{PR} = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2}, \quad \beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}}.$$

The corresponding methods are called FR (Fletcher-Revees), PR (Polak-Ribiére) and HS (Hestenes-Stiefel) conjugate gradient method respectively. Among them the PR method is regarded as the best one in practical computation. However, PR conjugate gradient method has no global convergence in many situations. Some modified PR conjugate gradient methods with global convergence were proposed (e.g., Grippo and Lucidi [9]; Shi [22], etc.). In PR conjugate gradient method, if $g_k^T d_k \geq 0$ occurs, we set $d_k = -g_k$ or set $d_k = -g_k$ at every n iteration. This is called restart conjugate gradient method (Powell [18]).

We tested our algorithms with a termination criterion when $\|g_k\| \leq eps$ with $eps = 10^{-8}$. The number of iterations used to get that precision is called IN. The number of function evaluations for getting the same error is denoted by FN.

We chose 18 test problems from the literature, including More, Garbow and Hillstrom, 1981, [13]; the BB gradient method combined with Raydan and Svaiter's CBB method [19]; and PR conjugate gradient algorithm with restart. The Raydan and Svaiter's CBB method is an efficient nonmonotone gradient

method [2, 20, 21], which is sometimes superior to some CG methods [2]. The initial iterative points were also from the literature [13].

For PR conjugate gradient method with restart, we use Wolfe rule (g) with $\beta = 0.75, \sigma = 0.125$. For our descent algorithms without line search, we choose the parameters $\delta = 1$, $M = 10^8$ and L_k defined by (44), (45), (46) and (47) respectively. The corresponding algorithms are denoted by A1, A2, A3 and A4 respectively.

Failures in the application of the new method may appear, mainly due to the inadequate estimations of L_k and sometimes due to the roundoff errors. In order to avoid failure, we check $f(x_k - \frac{\delta}{L_k} g_k) < f_k$ in our numerical experiment. It is obbserved that $\delta \in (0, 2)$ is an adjustable parameter in the gradient method without line search. We can adjust δ to satisfy the descent property of objective functions and improve the performance of the new method. If $f(x_k - \frac{\delta}{L_k} g_k) < f_k$ holds then we continue the iteration. Otherwise, we may reduce δ by setting $\delta = \gamma\delta$ with $\gamma \in (0, 1)$ until the descent property holds.

In numerical experiments, by Theorem 3.1, L_k may converge to $+\infty$. Thus, letting $L_k \in [L_0, M]$ seems to be unreasonable. In fact, we have no criteria to determine L_0 and M. Generally, a very large L_0 may lead to slow convergence rate and a very large M may violate the global convergence. As a result, we should choose carefully L_0 and M in practical computation in order to satisfy both the global convergence and the fast convergence rate. Actually, we can combine the step-size estimation and line search procedure to produce efficient descent algorithms. For example, in Armijo'e line search rule, $L > 0$ is a constant at each iteration, and we can take the initial step-size $s = s_k = 1/L_k$ at the k-th iteration. In this case, the steepest descent method has the same numerical performance as our corresponding descent algorithm. Accordingly, choosing an adequate initial step-size is very important for Armijo's line search and it is also the real aim of step-size estimation.

We use a Pentium IV portable computer and Visual C++ to implement our algorithms and test the 18 problems. The numerical results are reported in Table 1.

The computational results show that the new method in the paper is efficient in practice. It needs much less iterative number and less function evaluations and

P	n	A1	A2	A3	A4	PR	CBB
1	2	18	21	25	24	23,35	25
2	2	22	26	22	24	28,54	23
3	2	12	14	10	15	16,48	13
4	2	18	25	21	23	32,38	22
5	2	15	18	16	19	38,56	17
6	2	13	11	7	15	16,24	12
7	3	36	38	34	26	37,38	35
8	3	34	36	32	35	36,47	35
9	3	19	22	15	26	16,35	23
10	3	19	18	13	15	23,32	21
11	3	28	39	37	44	49,53	37
12	3	26	42	27	34	52,59	26
13	4	23	19	17	25	26,48	24
14	4	39	43	38	45	46,78	46
15	4	36	35	27	73	42,47	43
16	4	38	42	47	45	55,62	43
17	5	58	52	45	53	52,86	42
18	6	37	32	37	39	41,67	35
CPU	–	45	32	28	42	49	29

Table 1 – Numerical results of algorithms on $IN/FN(eps = 10^{-8})$.

then less CPU time (seconds) than PR conjugate gradient with restart and CBB method in some situations. This also shows that CBB method is a promising method because it is superior to the new algorithms A1, A2 and A4 for these test problems.

The gradient method with L_k defined by (45) seems to be the best algorithm for these test problems. Thereby, to estimate L_k is the key to constructing gradient methods without line search. If we take

$$\alpha_k = \frac{1}{L_k} = \frac{\|\delta_{k-1}\|^2}{\delta_{k-1}^T y_{k-1}},$$

or

$$\alpha_k = \frac{\delta_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2},$$

where $\delta_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$, we can obtain Barzilai and Borwein's method ([2]) which is an effective method for solving large scale unconstrained minimization problems.

Conclusion

In future research we should seek more approaches for estimating the step-size as exactly as possible and find some available technique to guarantee both the global convergence and quick convergence rate of gradient methods. We can also use some step estimation approaches to improve the original BB method and conjugate gradient methods, for example, [19].

Acknowledgements. The work was supported in part by NSF DMI-0514900, Postdoctoral Fund of China and K.C.Wong Postdoctoral Fund of CAS grant 6765700. The authors would like to thank the anonymous referees and the editor for many suggestions and comments.

REFERENCES

[1] L. Armijo, Minimization of function having Lipschitz continuous first partial derivatives, Pacific J. Math., **16** (1966), 1–13.

[2] J. Barzilai and J.M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal., **8** (1988), 141–148.

[3] E.G. Birgin and J. M. Martinez, A spectral conjugate gradient method for unconstrained optimization, Appl. Math. Optim., **43** (2001), 117–128.

[4] A. I. Cohen, Stepsize analysis for descent methods, J. Optim. Theory Appl., **33**(2) (1981), 187–205.

[5] H. B. Curry, The method of steepest descent for non-linear minimization problems, Quart. Appl. Math., **2** (1944), 258–261.

[6] Y. H. Dai and L. Z. Liao, R-linear convergence of the Barzilai and Borwein gradient method, IMA J. Numer. Anal., **22** (2002), 1–10.

[7] R. Fletcher, The Barzilai Borwein method-steepest descent method resurgent? Report in the International Workshop on "Optimization and Control with Applications", Erice, Italy, July (2001), 9–17.

[8] A. V. Fiacco, G. P. McCormick, Nonlinear programming: Sequential Unconstrained Minimization Techniques, SIAM, Philadelphia, (1990).

[9] L. Grippo and S.Lucidi, A globally convergent version of the Polak-Ribiere conjugate gradient, Math. Prog., **78** (1997), 375–391.

[10] A. A. Goldstein, Cauchy's method of minimization, Numer. Math. **4** (1962), 146–150.

[11] A. A. Goldstein, On steepest descent, SIAM J. Control, **3** (1965), 147–151.

[12] A. A. Goldstein, J. F. Price, An effective algorithm for minimization, Numer. Math., **10** (1967), 184–189.

[13] J. J. Moré, B. S. Garbow and K. E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software, **7** (1981), 17–41.

[14] J. Nocedal and J. S. Wright, Numerical Optimization, Springer-Verlag New York, Inc. (1999).

[15] J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numerica, **1** (1992), 199–242.

[16] M. J. D. Powell, Direct search algorithms for optimization calculations, Acta Numerica, **7** (1998), 287–336.

[17] E. Polak, Optimization: Algorithms and Consistent Approximations, Springer, New York, (1997).

[18] M. J. D. Powell, Restart procedure for the conjugate gradient method, Math. Prog., **12** (1977), 241–254.

[19] M. Raydan and B. F. Svaiter, Relaxed steepest descent and Cauchy-Barzilai- Borwein method, Comput. Optim. Appl., **21** (2002), 155–C167.

[20] M. Raydan, The Barzilai and Borwein method for the large scale unconstrained minimization problem, SIAM J. Optim., **7** (1997), 26–33.

[21] M. Raydan, On the Barzilai Borwein gradient choice of steplength for the gradient method, IMA J. Numer. Anal., **13** (1993), 321–326.

[22] Z. J. Shi, Restricted PR conjuate gradient method and its convergence(in Chinese), Advances in Mathematics, **31**(1) (2002), 47–55.

[23] P. Wolfe, Convergence condition for ascent methods II: some corrections, SIAM Rev., **13** (1971), 185–188.

[24] P. Wolfe, Convergence condition for ascent methods, SIAM Rev., **11** (1969), 226–235.

[25] Y. Yuan and W. Y. Sun, Optimization Theory and Methods, Science Press, Beijing (1997).

[26] Y. Yuan, Numerical Methods for Nonlinear Programming, Shanghai Scientific & Technical Publishers, (1993).