JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 127, No. 2, pp. 425–446, November 2005 (© 2005)
DOI: 10.1007/s10957-005-6553-6

New Inexact Line Search Method for Unconstrained Optimization[1,2]

Z. J. SHI[3] AND J. SHEN[4]

Communicated by F. Zirilli

Abstract. We propose a new inexact line search rule and analyze the global convergence and convergence rate of related descent methods. The new line search rule is similar to the Armijo line-search rule and contains it as a special case. We can choose a larger stepsize in each line-search procedure and maintain the global convergence of related line-search methods. This idea can make us design new line-search methods in some wider sense. In some special cases, the new descent method can reduce to the Barzilai and Borewein method. Numerical results show that the new line-search methods are efficient for solving unconstrained optimization problems.

Key Words. Unconstrained optimization, inexact line search, global convergence, convergence rate.

1. Introduction

Let R^n be an n-dimensional Euclidean space and let $f : R^n \to R^1$ be a continuously differentiable function. Line-search methods for solving the unconstrained minimization problem

$$\min f(x), \quad x \in R^n, \tag{1}$$

[1]The work was supported by NSF of China Grant 10171054, Postdoctoral Fund of China, and K. C. Wong Postdoctoral Fund of CAS Grant 6765700.
[2]The authors thank the anonymous referees for constructive comments and suggestions that greatly improved the paper.
[3]Professor, College of Operations Research and Management, Qufu Normal University, Rizhao, Shandong, PRC; Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, PRC.
[4]Assistant Professor, Department of Computer and Information Science, University of Michigan, Dearborn, Michigan.

have the form defined by the equation

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

where $x_1 \in R^n$ is an initial point, d_k is a descent direction of $f(x)$ at x_k, and α_k is the stepsize. Let x_k be the current iterative point, $k = 1, 2, 3, \ldots$, and x^* be a stationary point which satisfies $\nabla f(x^*) = 0$. We denote the gradient $\nabla f(x_k)$ by g_k, the function value $f(x_k)$ by f_k, and the function value $f(x^*)$ by f^*.

Choosing the search direction d_k and determining the stepsize α_k along the search direction at each iteration are the main tasks in line-search methods. The search direction d_k is generally required to satisfy

$$g_k^T d_k < 0, \tag{3}$$

which guarantees that d_k is a descent direction of $f(x)$ at x_k. In order to guarantee global convergence, we require sometimes that d_k satisfies the sufficient descent condition

$$g_k^T d_k \le -c \parallel g_k \parallel^2, \tag{4}$$

where $c > 0$ is a constant. Moreover, we need to choose d_k to satisfy the angle property

$$\cos < -g_k, d_k > = -g_k^T d_k / (\parallel g_k \parallel \cdot \parallel d_k \parallel) \ge \eta_0, \tag{5}$$

where $\eta_0 \in (0, 1]$ is a constant and $< -g_k, d_k >$ denotes the angle between the vectors $-g_k$ and d_k.

The commonly-used line-search rules are as follows.

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

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k). \tag{6}$$

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

$$\alpha_k = \min \left\{ \alpha | g(x_k + \alpha d_k)^T d_k = 0, \alpha > 0 \right\}. \tag{7}$$

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

$$f_k - f(x_k + \alpha d_k) \ge -\sigma \alpha g_k^T d_k. \tag{8}$$

(d) Limited Minimization Rule. Set $s_k = -g_k^T d_k/(L \parallel d_k \parallel)^2$. α_k is defined by

$$f(x_k+\alpha_k d_k) = \min_{\alpha\in[0,s_k]} f(x_k+\alpha d_k), \tag{9}$$

where $L>0$ is a constant.

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

$$\sigma \le [f(x_k+\alpha_k d_k) - f_k]/\alpha_k g_k^T d_k \le 1-\sigma. \tag{10}$$

It is possible to show that, if f is bounded below, there exists an interval of stepsize α_k for which the relation above is satisfied; there are fairly simple algorithms for finding such a stepsize through a finite number of arithmetic operations.

(f) Strong Wolfe Rule. α_k is chosen to satisfy simultaneously

$$f_k - f(x_k+\alpha_k d_k) \ge -\sigma\alpha_k g_k^T d_k, \tag{11}$$

$$|g(x_k+\alpha_k d_k)^T d_k| \le -\beta g_k^T d_k, \tag{12}$$

where σ and β are some scalars with $\sigma \in (0, 1/2)$ and $\beta \in (\sigma, 1)$.

(g) Wolfe Rule. α_k is chosen to satisfy (11) and

$$g(x_k+\alpha_k d_k)^T d_k \ge \beta g_k^T d_k. \tag{13}$$

Some important global convergence results for various methods using the above-mentioned specific line-search procedures have been given; see e.g. Refs. 1–7. In fact, the above-mentioned line-search methods are monotone descent for unconstrained optimization (Refs. 8–12). Nonmonotone line-search methods have been investigated also by many authors; see for example Refs. 13–15. In fact, the Barzilai-Borwein method (see Refs. 16–19) is a nonmonotone descent method which is an efficient algorithm for solving some special problems.

In this paper, we extend the Armijo line-search rule and analyze the global convergence of the corresponding descent methods. This new line-search rule is similar to the Armijo line-search rule and contains it as a special case. The new line-search rule can enable us to choose larger stepsize at each iteration and reduce the number of function evaluations at each step. This idea can make us design new line-search methods in some wider sense and find some new global convergence properties. In some special cases, the new descent method can reduce to the Barzilai and Borwein method (Refs. 16–19), which is regarded as an efficient algorithm for un-constrained optimization. Numerical results show that these new line-search methods are efficient for solving unconstrained optimization problems.

The paper is organized as follows. In section 2, we describe the new line-search rule. In sections 3 and 4, we analyze its convergence and convergence rate. In Section 5, we propose some ways to estimate the parameters used in the new line-search rule and report some numerical results. Conclusions are given in Section 6.

2. Inexact Line-Search Rule

Throughout the paper we make the following assumptions.

(H1) The function $f(x)$ has a lower bound on the level set $L_0 = \{x \in R^n \mid f(x) \leq f(x_0)\}$, where x_0 is given.

(H2) The gradient $g(x)$ of $f(x)$ is Lipschitz continuous in an open convex set B that contains L_0; i.e., there exists L such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in B.$$

We describe first the algorithm model.

Algorithm Model A.

Step 0. Given some parameters and the initial point x_1, set $k := 1$.
Step 1. If $\|g_k\| = 0$, then stop; else go to Step 2;
Step 2. Set $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction of $f(x)$ at x_k and where α_k is selected by some line-search rule.
Step 3. Set $k := k + 1$; go to Step 1.

In this section, we do not discuss how to choose d_k at each iteration, but investigate how to choose the stepsize α_k. Some useful line-search rules have been mentioned in the previous section. We describe here a new inexact line search rule which contains the Armijo line search rule as a special case. We will find that the stepsize defined in the new line-search rule is larger than that defined in the original Armijo line search rule. In other words, the stepsize defined by the new line-search rule is easier to find than that defined by the original Armijo line-search rule.

In particular, the stepsize defined in the modified Armijo line-search rule must be greater than the stepsize defined by the original Armijo line-search rule.

(c') Modified Armijo Line Search Rule. Set scalars s_k, β, L_k, μ, and σ with $s_k = -g_k^T d_k / L_k \|d_k\|^2$, $\beta \in (0, 1)$, $L_k > 0$, $\mu \in [0, 2)$, and

$\sigma \in (0, 1/2)$. Let α_k be the largest α in $\{s_k, \beta s_k, \beta^2 s_k, \ldots, \}$ such that

$$f(x_k + \alpha d_k) - f_k \leq \sigma\alpha[g_k^T d_k + (1/2)\alpha\mu L_k \|d_k\|^2]. \tag{14}$$

Remark 2.1. Suppose that α_k is defined by the line-search rule (c) and that α_k' is defined by the line-search rule (c'); then, $\alpha_k' \geq \alpha_k$. In other words, let A^c denote the set of the Armijo stepsize and let $A^{c'}$ be the set of new stepsizes; then, we have $A^c \subseteq A^{c'}$. In fact, if $L_k \equiv L$ and there exists α_k satisfying (8), then α_k is certain to satisfy (14).

Moreover, if $\mu = 0$, then the line search rule (c') reduces to the Armijo line search rule (c).

In Algorithm Model (A), the corresponding algorithms with line-search rule (c') is denoted by Algorithm (c'). In what follows, we analyze the global convergence of the new line-search method.

3. Global Convergence Analysis

Theorem 3.1. Assume that (H1) and (H2) hold, the search direction d_k satisfies (3), and α_k is determined by the modified Armijo line-search rule. Algorithm (e') generates an infinite sequence $\{x_k\}$ and

$$0 < L_k \leq m_k L, \tag{15}$$

where m_k is a positive integer and $m_k \leq M_0 < +\infty$, with M_0 being a large positive constant. Then,

$$\sum_{k=1}^{\infty} \left(g_k^T d_k / \|d_k\|\right)^2 < +\infty. \tag{16}$$

Proof. Let

$$K_1 = \{k | \alpha_k = s_k\}, \quad K_2 = \{k | \alpha_k < s_k\}.$$

If $k \in K_1$, then

$$\begin{aligned} f(x_k + \alpha_k d_k) - f_k &\leq \sigma\alpha_k[g_k^T d_k + (1/2)\alpha_k L_k \|d_k\|^2] \\ &= -\sigma \left[g_k^T d_k / L_k \|d_k\|^2\right] \left[g_k^T d_k - (1/2)\mu g_k^T d_k\right] \\ &= -[\sigma(1 - (1/2)\mu)/L_k] \left(g_k^T d_k\right)^2 / \|d_k\|^2; \end{aligned}$$

thus,

$$f(x_k+\alpha_k d_k)-f_k \le -[\sigma(1-(1/2)\mu)/L_k]\left(g_k^T d_k\right)^2/\|d_k\|^2, \quad k\in K_1. \quad (17)$$

Let

$$\eta_k=-\sigma(1-(1/2)\mu)/L_k, \quad k\in K_1;$$

by (15), we have

$$\begin{aligned}\eta_k &= -\sigma(1-(1/2)\mu)/L_k\\ &\le -\sigma(1-(1/2)\mu)/m_k L\\ &\le -\sigma(1-(1/2)\mu)/M_0 L<0.\end{aligned}$$

Let

$$\eta'=\sigma(1-(1/2)\mu)/M_0 L.$$

This and (17) imply that $\eta_k\le-\eta'$ and

$$f_{k+1}-f_k\le-\eta'\left(g_k^T d_k/\|d_k\|\right)^2, \quad k\in K_1. \quad (18)$$

If $k\in K_2$, then $\alpha_k<s_k$. This shows that s_k cannot satisfy (14) and thus $\alpha_k\le \beta s_k$. By the modified Armijo line-search rule (e'), we assert that $\alpha=\alpha_k/\beta$ cannot satisfy (14) and thus

$$f(x_k+\alpha_k d_k/\beta)-f_k>\sigma\alpha_k/\beta[g_k^T d_k+(1/2)\alpha_k\mu L_k\|d_k\|^2/\beta].$$

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

$$\alpha_k g(x_k+\theta_k\alpha_k d_k/\beta)^T d_k/\beta>\sigma\alpha_k/\beta[g_k^T d_k+(1/2)\alpha_k\mu L_k\|d_k\|^2/\beta];$$

therefore,

$$g(x_k+\theta_k\alpha_k d_k/\beta)^T d_k>\sigma\left[g_k^T d_k+(1/2)\alpha_k\mu L_k\|d_k\|^2/\beta\right]. \quad (19)$$

By (H2), the Cauchy-Schawarz inequality, and (19), we obtain

$$\begin{aligned}\alpha_k L\|d_k\|^2/\beta &\ge \|g(x_k+\theta_k\alpha_k d_k/\beta)-g_k\|\cdot\|d_k\|\\ &\ge [g(x_k+\theta_k\alpha_k d_k/\beta)-g_k]^T d_k\\ &> -(1-\sigma)g_k^T d_k+(1/2)\sigma\mu\alpha_k L_k\|d_k\|^2/\beta.\end{aligned}$$

Therefore,

$$\alpha_k L\|d_k\|^2/\beta > -(1-\sigma)g_k^T d_k,$$

which implies that

$$\alpha_k \geq -[\beta(1-\sigma)/L]\, g_k^T d_k/\|d_k\|^2, \quad k \in K_2. \tag{20}$$

Letting

$$s_k' = -[\beta(1-\sigma)/L]\, g_k^T d_k/\|d_k\|^2, \quad k \in K_2,$$

we have

$$s_k > \alpha_k > s_k', \quad k \in K_2. \tag{21}$$

By (14) and (21), we have

$$\begin{aligned} f_{k+1} - f_k &\leq \sigma\alpha_k \left[g_k^T d_k + (1/2)\alpha_k \mu L_k \|d_k\|^2\right] \\ &\leq \sigma \max_{s_k \geq \alpha \geq s_k'} \left\{\alpha \left[g_k^T d_k + (1/2)\alpha\mu L_k \|d_k\|^2\right]\right\} \\ &\leq \sigma \max_{s_k \geq \alpha \geq s_k'} \left\{\alpha \left[g_k^T d_k + (1/2)s_k\mu L_k \|d_k\|^2\right]\right\} \\ &= \sigma s_k'(1-(1/2)\mu) g_k^T d_k \\ &= -[\sigma\beta(1-\sigma)(1-(1/2)\mu)/L]\left(g_k^T d_k/\|d_k\|\right)^2. \end{aligned}$$

Letting

$$\eta'' = \sigma\beta(1-\sigma)(1-(1/2)\mu)/L, \tag{22}$$

we have

$$f_{k+1} - f_k \leq -\eta''(g_k^T d_k)^2/\|d_k\|^2, \quad k \in K_2. \tag{23}$$

Let

$$\eta_0' = \min(\eta', \eta'');$$

by (18) and (23), we have

$$f_{k+1} - f_k \leq -\eta_0' \left(g_k^T d_k/\|d_k\|\right)^2, \quad \forall k. \tag{24}$$

By (H1) and (24), we can obtain that $\{f_k\}$ is a decreasing sequence and has a bound from below. This shows that $\{f_k\}$ has a limit. By (24), we prove via (H1) that (16) holds. □

Corollary 3.1. If the conditions in Theorem 3.1 hold, then

$$\lim_{k \leftarrow \infty} \left(g_k^T d_k / \|d_k\| \right)^2 = 0. \tag{25}$$

In fact, Assumption H2 can be replaced by the following weaker assumption:

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

Theorem 3.2. Assume that (H1) and (H2′) hold, the search direction d_k satisfies (3), and α_k is determined by the modified Armijo line-search rule. Algorithm (e') generates an infinite sequence $\{x_k\}$ and

$$0 < L_k \le M_0', \tag{26}$$

where M_0' is a large positive constant. Then,

$$\lim_{k \to \infty} \left(-g_k^T d_k / \|d_k\| \right) = 0. \tag{27}$$

Proof. Similarly as in the proof of Theorem 3.1, if $k \in K_1$, by (18), we can prove that

$$\lim_{k \in K_1, k \to \infty} \left(-g_k^T d_k / \|d_k\| \right) = 0. \tag{28}$$

In the case of $k \in K_2$, by (14), we have

$$\begin{aligned} f(x_k + \alpha_k d_k) - f_k &\le \sigma \alpha_k [g_k^T d_k + (1/2)\alpha_k \mu L_k \|d_k\|^2] \\ &\le \sigma \alpha_k [g_k^T d_k + (1/2) s_k \mu L_k \|d_k\|^2] \\ &= \sigma \alpha_k (1 - (1/2)\mu) g_k^T d_k. \end{aligned}$$

By (H1), we have

$$\lim_{k \in K_2,\ k \to \infty} \left(-\alpha_k g_k^T d_k \right) = 0. \tag{29}$$

If there exist $\epsilon_0 > 0$ and an infinite subset $K_3 \subseteq K_2$ such that

$$-g_k^T d_k / \|d_k\| \ge \epsilon_0, \quad \forall k \in K_3, \tag{30}$$

then by (29) and (30), we have

$$\lim_{k\in K_3,\ k\to\infty} \alpha_k \|d_k\| = 0. \tag{31}$$

By (19), we have

$$g(x_k + \theta_k \alpha_k d_k/\beta)^T d_k \geq \sigma g_k^T d_k, \quad k \in K_3, \tag{32}$$

where $\theta_k \in [0, 1]$ is defined in the proof of Theorem 3.1. By the Cauchy-Schwarz inequality and (32), we have

$$\begin{aligned}\|g(x_k + \theta_k\alpha_k d_k/\beta) - g_k\| &= \|g(x_k + \theta_k\alpha_k d_k/\beta) - g_k\| \|d_k\|/\|d_k\| \\ &\geq [g(x_k + \theta_k\alpha_k d_k/\beta) - g_k]^T d_k/\|d_k\| \\ &\geq -(1-\sigma) g_k^T d_k/\|d_k\|, \quad k \in K_3.\end{aligned}$$

By (H2′) and (31), we obtain

$$\lim_{k\in K_3,\ k\to\infty} \left(-g_k^T d_k/\|d_k\|\right) = 0,$$

which contradicts (30). This shows that

$$\lim_{k\in K_2,\ k\to\infty} \left(-g_k^T d_k/\|d_k\|\right) = 0. \tag{33}$$

By (28), (33), and noting that $K_1 \bigcup K_2 = \{1, 2, 3 \ldots\}$, we assert that (27) holds. □

Since (H2) implies (H2′), Theorem 3.1 is essentially a corollary of Theorem 3.2.

4. Linear Convergence Rate

In order to analyze the convergence rate, we assume that the sequence $\{x_k\}$ generated by the new algorithm converges to x^*. We make further the following assumption.

(H3) $\nabla^2 f(x^*)$ is a symmetric positive-definite matrix and $f(x)$ is twice continuously differentiable on a neighborhood $N_0(x^*, \epsilon_0)$ of x^*.

Lemma 4.1. Assume that (H3) holds. Then, there exist $\epsilon > 0$ and $0 < m' \leq M'$ such that (H1) and (H2) hold for $x_0 \in N(x^*, \epsilon) \subseteq N_0(x^*, \epsilon_0)$ and

$$m'\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M'\|y\|^2, \quad \forall x, y \in N(x^*, \epsilon), \tag{34}$$

$$(1/2)m'\|x - x^*\| \leq f(x) - f(x^*) \leq (1/2)M'\|x - x^*\|^2, \quad \forall x \in N(x^*, \epsilon), \tag{35}$$

$$M'\|x - y\|^2 \geq (g(x) - g(y))^T (x - y) \geq m'\|x - y\|^2, \quad \forall x, y \in N(x^*, \epsilon). \tag{36}$$

Thus,

$$M'\|x - x^*\|^2 \geq g(x)^T (x - x^*) \geq m'\|x - x^*\|^2, \quad \forall x \in N(x^*, \epsilon). \tag{37}$$

By (37) and (36), we obtain also from the Cauchy-Schwarz inequality that

$$M'\|x - x^*\| \geq \|g(x)\| \geq m'\|x - x^*\|, \quad \forall x \in N(x^*, \epsilon), \tag{38}$$

and

$$\|g(x) - g(y)\| \leq M'\|x - y\|, \quad \forall x, y \in N(x^*, \epsilon). \tag{39}$$

Its proof can be seen from e.g. Refs. 7–8 or Refs. 21–22.

Lemma 4.2. If (H1) and (H2) hold, the search direction d_k satisfies the angle property (5) at each iteration, Algorithm (c') generates an infinite sequence $\{x_k\}$; then, there exists $\eta > 0$ such that

$$f_k - f_{k+1} \geq \eta\|g_k\|^2, \quad \forall k. \tag{40}$$

Proof. By Theorem 3.1, (5), and (24), we have

$$\begin{aligned} f_{k+1} - f_k &\leq -\eta_0' \left(g_k^T d_k / \|d_k\|\right)^2 \\ &= -\eta_0' \left(g_k^T d_k / \|g_k\| \cdot \|d_k\|\right)^2 \cdot \|g_k\|^2 \\ &\leq -\eta_0' \eta_0^2 \|g_k\|^2. \end{aligned}$$

Let

$$\eta = \eta_0' \eta_0^2;$$

we obtain that (40) holds. □

Theorem 4.1. If (H3) holds, the search direction d_k satisfies the angle property (5) at each iteration, Algorithm (c') generates an infinite sequence $\{x_k\}$, and $x_k \in N(x^*, \epsilon)$ for sufficiently large k. Then, $\{x_k\} \to x^*$ at least linearly.

Proof. By (H3), Lemma 4.1, Lemma 4.2, and (38), it follows that

$$\lim_{k\to\infty} x_k = x^*.$$

By (38) and Lemma 4.1, we obtain

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

Setting

$$\theta = m'\sqrt{2\eta/M'},$$

we can prove that $\theta < 1$. In fact, by the definition of η and noting that $M' \leq L$, we obtain

$$\begin{aligned} \theta^2 &= 2m'^2\eta/M' \leq 2m'^2\eta_0^2\eta_0'/M' \leq 2m'^2\eta_0'/M' \\ &\leq 2m'^2\eta'/M' \leq [\sigma(1-(1/2)\mu)/M_0 L]\left(2m'^2/M'\right) \\ &= [2\sigma(1-(1/2)\mu/M_0]\, m'^2/M'^2 \leq (2\sigma/M_0) < 1. \end{aligned}$$

Set

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

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

$$\begin{aligned} f_{k+1} - f^* &\leq (1-\theta^2)(f_k - f^*) \\ &= \omega^2(f_k - f^*) \\ &\leq \cdots \\ &\leq \omega^{2k}(f_1 - f^*). \end{aligned}$$

By Lemma 4.1 and the above inequality, we have

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq (2/m')(f_{k+1} - f^*) \\ &\leq \omega^{2k}\left[2(f_1 - f^*)/m'\right]; \end{aligned}$$

thus,

$$\|x_{k+1}-x^*\| \le \omega^k \sqrt{2(f_1-f^*)/m'};$$

therefore,

$$\begin{aligned} R_1\{x_k\} &= \lim_{k\to+\infty} \|x_k - x^*\|^{1/k} \\ &= \lim_{k\to+\infty} \left[\omega^{k-1}\sqrt{2(f_1-f^*)/m'}\right]^{1/k} \\ &= \omega \lim_{k\to+\infty} \left[\sqrt{2(f_1-f^*)/m'}\right]^{1/k} \\ &= \omega < 1, \end{aligned}$$

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

5. Numerical Results

In this section, we discuss the implementation of the new algorithm. The technique of choosing parameters is reasonable and effective for solving practical problems in both theory and numerical aspects.

5.1. Parameter Estimation. In the modified Armijo line-search rule, there is a parameter L_k which must be estimated. As we know, L_k should approximate the Lipschitz constant M' of the gradient $g(x)$ of the objective function $f(x)$. If M' is given, we should certainly take $L_k = M'$. However, M' is not known prior in many situations. L_k needs to be estimated in some cases.

First of all, let

$$\delta_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = g_k - g_{k-1}, \quad k = 2, 3, 4, \dots,$$

and estimate

$$L_k = \|y_{k-1}\| / \|\delta_{k-1}\| \tag{41}$$

or

$$L_k = \max\{\|y_{k-i}\| / \|\delta_{k-i}\| \,|\, i = 1, 2, \dots, M\}, \tag{42}$$

whenever $k \ge M+1$, where M is a positive integer.

Next, the BB method (Refs. 16–19) motivates us also to find a way of estimating M'. Here, BB stands for Bazzilai and Borwein. Solving the minimization

$$\min \|L_k \delta_{k-1} - y_{k-1}\|,$$

we obtain

$$L_k = \delta_{k-1}^T y_{k-1} / \|\delta_{k-1}\|^2. \tag{43}$$

Obviously, if $k \geq M+1$, we can also take

$$L_k = \max\left\{\delta_{k-i}^T y_{k-i} / \|\delta_{k-i}\|^2|, \quad i = 1, 2, \ldots, M\right\}. \tag{44}$$

On the other hand, we can take

$$L_k = \|y_{k-1}\|^2 / \delta_{k-1}^T y_{k-1} \tag{45}$$

or

$$L_k = \max\left\{\|y_{k-i}\|^2 / \delta_{k-i}^T y_{k-i}| \quad i = 1, 2, \ldots, M\right\}, \tag{46}$$

whenever $k \geq M+1$.

There are many other techniques of estimating the Lipschitz constant M'; see Ref. 8. We will use (41)–(46) to estimate M' and the corresponding algorithms are denoted as Algorithms (41)–(46) respectively.

5.2. Numerical Results. In what follows, we will discuss the numerical performance of the new line-search method. The test problems are chosen from Ref. 20 and the implementable algorithm is stated as follows.

Algorithm A.

Step 0. Given some parameters $\sigma \in (0, 1/2)$, $\beta \in (0, 1)$, $\mu \in [0, 2)$, and $L_1 = 1$, let $x_1 \in R^n$ and set $k := 1$.

Step 1. If $\|g_k\| = 0$, then stop; else, go to Step 3.

Step 3. Choose d_k to satisfy the angle property (5); for example, choose $d_k = -g_k$.

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$, where α_k is defined by the modified Armijo line search rule.

Step 5. Set $\delta_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, and L_{k+1} is determined by one of (41)–(46).

Step 6. Set $k := k+1$; go to Step 1.

Table 1. Iterations and function evaluations, $\mu = 1$.

P	n	Armijo	New (41)	New (43)	New (45)
P5	2	8, 12	6, 9	7, 11	7, 8
P13	4	25, 38	18, 22	16, 25	17, 26
P14	4	36, 50	28, 34	26, 33	30, 42
P16	4	14, 72	16, 63	12, 58	11, 56
P20	9	12, 17	12, 13	12, 13	11, 11
P21	16	18, 21	16, 23	12, 14	11, 15
P21	100	21, 30	17, 22	16, 25	15, 20
P23	8	30, 42	28, 34	26, 36	26, 38
P23	100	36, 58	30, 34	28, 32	30, 52
P23	200	55, 87	43, 67	48, 72	42, 61
P24	20	52, 67	45, 59	38, 38	47, 52
P25	50	11, 121	11, 32	16, 78	9, 83
P26	50	14, 30	14, 19	12, 16	15, 18
P30	20	13, 22	11, 18	10, 19	12, 19

In the above algorithm, we set

$$\sigma = 0.38, \quad \beta = 0.87, \quad \mu = 1$$

and set the same parametric values in the original Armijo line-search method with $L = 1$. We will find that the stepsize in the new line-search method is easier to find than in the original one. In other words, the new line-search method needs less evaluations of gradients and objective functions at each iteration. We tested the new line-search methods and the original Armijo line-search method with double precision in a portable computer. The codes were written in the visual C++ language. Our test problems and the initial points used are drawn from Ref. 20. For each problem, the limiting number of function evaluations is set to 10000 and the stopping condition is

$$\|g_k\| \leq 10^{-6}. \tag{47}$$

Our numerical results are shown in Tables 1–4, where Armijo, New (41), New (43), New (45) stand for the Armijo line-search method and the new line search methods with L_k given by (41), (43), (45) respectively. The symbols n, In, N_f mean respectively the dimension of problems, the number of iterations, and the number of function evaluations, respectively.

The unconstrained optimization problems are numbered in the same way as in Ref. 8. For example, P5 means Problem 5 in Ref. 20.

Table 2. Iterations and function evaluations, $\mu = 1.5$.

P	n	Armijo	New (41)	New (43)	New (45)
P5	2	8, 12	6, 7	7, 10	6, 8
P13	4	25, 38	17, 20	16, 21	15, 23
P14	4	36, 50	23, 28	24, 31	25, 36
P16	4	14, 72	16, 43	12, 49	11, 38
P20	9	12, 17	12, 13	12, 13	11, 11
P21	16	18, 21	16, 21	11, 13	11, 14
P21	100	21, 30	16, 18	16, 22	15, 18
P23	8	30, 42	25, 33	26, 32	23, 32
P23	100	36, 58	28, 33	26, 28	30, 47
P23	200	55, 87	40, 58	42, 61	38, 56
P24	20	52, 67	43, 48	36, 38	42, 48
P25	50	11, 121	11, 28	16, 43	9, 67
P26	50	14, 30	14, 19	12, 16	15, 18
P30	20	13, 22	11, 16	10, 16	12, 17

Table 3. Iterations and function evaluations, $\mu = 1$.

P	n	Armijo	New (41)	New (43)	New (45)
P21	1000	98, 562	66, 320	58, 187	63, 213
P21	5000	143, 736	74, 421	87, 325	82, 288
P23	1000	120, 984	93, 437	78, 529	84, 512
P23	5000	185, 2842	126, 933	126, 922	113, 847
P23	8000	224, 3827	140, 1250	123, 1541	118, 1628
P24	5000	283, 6250	186, 4212	236, 3238	178, 2694
P25	5000	217, 8364	158, 2472	154, 3312	126, 3269
P26	5000	163, 1923	112, 1283	125, 1538	105, 1163
P30	5000	149, 926	121, 612	119, 583	108, 581

Table 4. Iterations and function evaluations, $\mu = 1.5$.

P	n	Armijo	New (41)	New (43)	New (45)
P21	1000	98, 562	58, 274	53, 148	63, 162
P21	5000	143, 736	68, 317	62, 236	78, 242
P23	1000	120, 984	82, 329	78, 412	81, 468
P23	5000	185, 2842	118, 726	118, 821	103, 687
P23	8000	224, 3827	132, 984	112, 965	116, 1263
P24	5000	283, 6250	147, 2872	182, 2893	158, 2476
P25	5000	217, 8364	127, 1963	128, 2305	114, 2129
P26	5000	163, 1923	112, 1132	126, 1259	105, 982
P30	5000	149, 926	98, 263	103, 321	89, 283

(P5) Beale Function. Here,

$$f(x)=\sum_{i=1}^{3} f_i(x)^2,$$
$$f_i(x)=y_i - x_1(1-x_2^i),\ \ i=1,2,3,$$
$$y_1=1.5,\ \ y_2=2.25,\ \ y_3=2.625.$$
$$x^1=(1,1)^T,\ \ x^*=(3,0.5)^T,\ \ f^*=0.$$

(P13) Powell Singular Function. Here,

$$f(x)=\sum_{i=1}^{4} f_i(x)^2,$$
$$f_1(x)=x_1+10x_2,\quad f_2(x)=5^{1/2}(x_3-x_4),$$
$$f_3(x)=(x_2-2x_3)^2,\quad f_4(x)=10^{1/2}(x_1-x_4)^4.$$
$$x^1=(3,-1,0,1)^T,\quad x^*=(0,0,0,0)^T,\quad f^*=0.$$

(P14) Wood Function. Here,

$$f(x)=\sum_{i=1}^{6} f_i(x)^2,$$
$$f_1(x)=10(x_2-x_1^2),\ \ f_2(x)=1-x_1,$$
$$f_3(x)=(90)^{1/2}(x_4-x_3^2),\ \ f_4(x)=1-x_3,$$
$$f_5(x)=(10)^{1/2}(x_2+x_4-2),\ \ f_6(x)=(10)^{-1/2}(x_2-x_4).$$
$$x^1=(-3,-1,-3,-1)^T,\ \ x^*=(0,0,0,0)^T,\ \ f^*=0.$$

(P16) Brown and Dennis Function. Here,

$$f(x)=\sum_{i=1}^{20} f_i(x)^2,$$
$$f_i(x)=[x_1+t_ix_2-\exp(t_i)]^2+[x_3+x_4\sin(t_i)-\cos(t_i)]^2,$$
$$t_i=i/5,\ \ \mathrm{i}=1,2,\dots,20.$$
$$x^1=(25,5,-5,-1)^T,\ \ f^*=85822.2\dots.$$

(P20) Watson Function. Here,

$$f(x)=\sum_{i=1}^{31} f_i(x)^2,$$
$$f_i(x)=\sum_{j=2}^{n}(j-1)x_j t_i^{j-2}-\left(\sum_{j=1}^{n} x_j t_i^{j-1}\right)^2-1,$$
$$t_i=i/29,\quad 1\le i\le 29,$$
$$f_{30}(x)=x_1,\quad f_{31}(x)=x_2-x_1^2-1.$$
$$x^1=(0,\dots,0)^T,\quad f^*=1.39976\cdots 10^{-6}.$$

(P21) Extended Rosenbrock Function. Here,

$$f(x)=\sum_{i=1}^{n} f_i(x)^2,$$
$$f_{2i-1}(x)=10(x_{2i}-x_{i-1}^2),\quad f_{2i}(x)=1-x_{2i-1}.$$
$$x^1=(\xi_j),\quad \xi_{2j-1}=-1.2,\quad \xi_{2j}=1.$$
$$x^*=(1,\dots,1)^T,\quad f^*=0.$$

(P23) Penalty Function I. Here,

$$f(x)=\sum_{i=1}^{n+1} f_i(x)^2,$$
$$f_i(x)=a^{1/2}(x_i-1),\quad 1\le i\le n,$$
$$f_{n+1}(x)=\left(\sum_{j=1}^{n} x_j^2\right)-1/4,\quad a=10^{-5}.$$
$$x^1=(\xi_j),\quad \xi_j=j.$$

(P24) Penalty Function II. Here,

$$f(x)=\sum_{i=1}^{2n} f_i(x)^2,$$
$$f_1(x)=x_1-2,$$
$$f_i(x)=a^{1/2}[\exp(x_i/10)+\exp(x_{i-1}/10-y_i],\quad 2\le i\le n,$$
$$f_i(x)=a^{1/2}[\exp(x_{i-n+1}/10)-\exp(-1/10)],\quad n<i<2n,$$
$$f_{2n}(x)=\left[\sum_{j=1}^{n}(n-j+1)x_j^2\right]-1,$$
$$a=10^{-5},\ \ y_i=\exp(i/10)+\exp[(i-1)/10].$$
$$x^1=(1/2,\ldots,1/2)^T.$$

(P25) Variably-Dimensioned Function. Here,

$$f(x)=\sum_{i=1}^{n+2} f_i(x)^2,$$
$$f_i(x)=x_i-1,\quad i=1,\ldots,n,$$
$$f_{n+1}(x)=\sum_{j=1}^{n} j(x_j-1),$$
$$f_{n+2}(x)=\left[\sum_{j=1}^{n} j(x_j-1)\right]^2.$$
$$x^1=(\xi_j),\quad \xi_j=1-(j/n).$$
$$x^*=(1,\ldots,1)^T,\quad f^*=0.$$

(P26) Trigonometric Function. Here,

$$f(x)=\sum_{i=1}^{n} f_i(x)^2,$$
$$f_i(x)=n-\sum_{j=1}^{n}\cos x_j+i(1-\cos x_i)-\sin x_i,\quad i=1,2,\ldots,n.$$
$$x^1=(1/n,\ldots,1/n)^T,\quad f^*=0.$$

(P30) Broyden Tridiagonal Function. Here,

$$f(x) = \sum_{i=1}^{n} f_i(x)^2,$$
$$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \quad i = 1, \ldots, n,$$
$$x_0 = x_{n+1} = 0.$$
$$x^1 = (-1, \ldots, -1)^T, \quad f^* = 0.$$

We set $d_k = -g_k$ at each iteration. In this case, if $\alpha_k = s_k$ at each iteration, then New (43) and New (45) will reduce to the BB methods (Refs. 16–19), because of

$$\alpha_k = s_k = -g_k^T d_k / L_k \|d_k\|^2 = 1/L_k$$
$$= \|\delta_{k-1}\|^2 / \delta_{k-1}^T y_{k-1},$$

corresponding to (43), or because of

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

corresponding to (45).

This shows that the new line-search method contains the BB methods and has global convergence, while the BB method has no global convergence in some cases. Thus, the new method is promising and will challenge the BB method in some sense.

In Table 1, a pair of numbers means that the first number denotes the number of iterations and the second number denotes the number of function evaluations when reaching the same precision (47), that is, the pair of numbers is (In, N_f). It can be seen that, for some problems, the three new algorithms need less number N_f of function evaluations than the original Armijo line-search method; On the other hand, for some problems, the Armijo line-search method performs as well as the new line-search method. Overall, our numerical results indicate that the new line-search methods are superior to the original Armijo line-search method in many situations. In particular, the new method needs less function evaluations than the original Armijo line-search method when reaching the same precision; i.e., in many cases, α_k often takes s_k in the new line-search rule. Moreover, the estimation of L_k and thus s_k is very important in the new algorithm. In the numerical experiment, we take $d_k = -g_k$, which is a very special case. We can take other descent directions as d_k at each step.

In the new line search method, there are some parameters $\sigma \in (0, 1/2)$, $\beta \in (0, 1)$, $\mu \in [0, 2)$ which need to be set in concrete algorithms. In practical computation, μ is a key parameter because the new method performs better when it increases in [0,2). The numerical results in Table 2 show the fact.

It is obvious that the numerical results in Table 2 are better than those in Table 1. The reason is that, when μ increases, s_k is easily accepted as the α_k and the function evaluations will decrease at each iteration. The results of Tables 1–2 show that the new algorithm works better than the original Armijo algorithm when the dimension n of the problem increases. We can increase the dimension n of some problems and conduct many more numerical experiments.

It can be seen from Tables 3 and 4 that the three new algorithms have better numerical performance than the original Armijo algorithm. When the dimension n of the problems increases, the new method works better than the original Armijo line search method. We can also see that, the greater $\mu \in [0, 2)$ is, the better the new method performs.

6. Conclusions

A new line-search rule is proposed and the related descent method is investigated. We can choose a larger stepsize in each line-search procedure and maintain the global convergence of related line-search method. This idea can make us design new line-search methods in some wider sense. Especially, the new method can reduce to BB method (Refs. 16–19) in some special cases. As we can see, the Lipschitz constant M' of the gradient $g(x)$ of the objective function $f(x)$ needs to be estimated at each step. We have discussed some techniques for choosing L_k. In the numerical experiment, we take $d_k = -g_k$ at each step. Indeed, we can take other descent directions as d_k.

For further research, we can establish other similar line-search rules such as the Goldstein rule and Wolfe rule. Of course, we hope in less evaluation numbers of the gradients and objective functions at each iteration. Since the new line-search rule needs to estimate L_k, we can find other ways to estimate L_k and choose diverse parameters such as σ, μ, β so as to find available parameters in solving special unconstrained optimization problems.

References

1. Dennis, J. E., and Schnable, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Ennglewood Cliffs, New Jersey, 1983.

2. Nocedal, J., *Theory of Algorithms for Unconstrained Optimization*, Acta Numerica, Vol. 1, pp. 199–242, 1992.
3. Powell, M. J. D., *Direct Search Algorithms for Optimization Calculations*, Acta Numerica, Vol. 7, pp. 287–336, 1998.
4. Wolfe, P., *Convergence Conditions for Ascent Methods*, SIAM Review, Vol. 11, pp. 226–235, 1969.
5. Wolfe, P., *Convergence Conditions for Ascent Methods, II: Some Corrections*, SIAM Review, Vol. 13, pp. 185–188, 1971.
6. Shi, J. Z., *Convergence of Line-Search Methods for Unconstrained Optimization*, Applied Mathematics and Computation, Vol. 157, pp. 393–405, 2004.
7. Cohen, A. I., *Stepsize Analysis for Descent Methods*, Journal of Optimizaiton Theory and Applications, Vol. 33, pp. 187–205, 1981.
8. Yuan, Y., *Numerical Methods for Nonlinear Programming*, Shanghai Scientific and Technical Publishers, 1993.
9. Todd, M. J., *On Convergence Properties of Algorithms for Unconstrained Minimization*, IMA Journal of Numerical Analysis, Vol. 9, pp. 435–441, 1989.
10. Shi, Z. J., *A Supermemory Gradient Method for Unconstrained Optimization Problems*, Chinese Journal of Engineering Mathematics, Vol. 17, pp. 99–104, 2000 (in Chinese).
11. Wei, Z., Qi, L., and Jiang, H., *Some Convergence Properties of Descent Methods*, Journal of Optimization Theory and Applications, Vol. 95, pp. 177–188, 1997.
12. Vrahatis, M. N., Androulakis, G. S., and Manoussakis, G. E., *A New Unconstrained Optimization Method for Imprecise Function and Gradient Values*, Journal of Mathematical Analysis and Applications, Vol. 197, pp. 586–607, 1996.
13. Grippo, L., Lampariello, F., and Lucidi, S., *A Class of Nonmonotone Stability Methods in Unconstrained Optimization*, Numerische Mathematik, Vol. 62, pp. 779–805, 1991.
14. Dai, Y. H., *On the Nonmonotone Line Search*, Journal of Optimization Theory and Applications, Vol. 112, pp. 315–330, 2002.
15. Sun, W. Y., Han, J. Y., and Sun, J., *Global Convergence of Nonmonotone Descent Methods for Unconstrained Optimization Problems*, Journal of Computational and Applied Mathematics, Vol. 146, pp. 89–98, 2002.
16. Barzilai, J., and Borwein, J. M., *Two-Point Stepsize Gradient Methods*, IMA Journal of Numerical Analysis, Vol. 8, pp. 141–148, 1988.
17. Dai, Y. H., and Liao, L. Z., *R-Linear Convergence of the Barzilai and Borwein Gradient Method*, IMA Journal of Numerical Analysis, Vol. 22, pp. 1–10, 2002.
18. Raydan, M., *The Barzilai and Borwein Method for the Large-Scale Unconstrained Minimization Problem*, SIAM Journal on Optimization, Vol. 7, pp. 26–33, 1997.
19. Raydan, M., *On the Barzilai-Borwein Gradient Choice of Steplength for the Gradient Method*, IMA Journal of Numerical Analysis, Vol. 13, pp. 321–326, 1993.

20. Moré, J. J., Garbow, B. S., and Hillstron, K. E., *Testing Unconstrained Optimization Software*, ACM Transactions on Mathematical Software, Vol. 7, pp. 17–41, 1981.
21. Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer Verlag, New York, NY, 1999.
22. Polak, E., *Optimization: Algorithms and Consistent Approximations*, Springer Verlag, New York, NY, 1997.