Computers and Mathematics with Applications 60 (2010) 2517–2525

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Nonmonotone second-order Wolfe's line search method for unconstrained optimization problems[☆]

Xue Han[a], Wenyu Sun[a,*], Chuangyin Dang[b]

[a] *School of Mathematical Sciences, Nanjing Normal University, Nanjing 210046, China*

[b] *Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong, China*

ARTICLE INFO

Article history:
Received 9 February 2010
Received in revised form 2 August 2010
Accepted 2 August 2010

Keywords:
Nonmonotone
Second-order line search
Descent pair
Wolfe's line search
Unconstrained optimization

ABSTRACT

In this paper, we present a new algorithm using the nonmonotone second-order Wolfe's line search. By using the negative curvature information from the Hessian, we prove that the generated sequence converges to the stationary points that satisfy the second-order optimality conditions. We also report numerical results which show the efficiency and robustness of the proposed method.

1. Introduction

In this paper, we consider the following unconstrained optimization problem:

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

Assume that $f : R^n \to R$ is a twice continuously differentiable function, both the gradient $g(x) = \nabla f(x)$ and the Hessian matrix $H(x) = \nabla^2 f(x)$ of f exist and are continuous. For the sake of simplicity, we will abbreviate $f(x_k)$, $g(x_k)$, and $H(x_k)$ as f_k, g_k and H_k, respectively. In addition, the notation $\|\cdot\|$ denotes the Euclidean norm on R^n, and $\lambda_{min}(\cdot)$ stands for the minimal eigenvalue of a matrix.

There are many iterative algorithms for solving the problem (1.1). They usually first find a descent direction, and then find a suitable step size along the direction. The key idea of these algorithms is to converge to a stationary point x^* where the Hessian matrix $H(x^*)$ is a positive semidefinite matrix. However, the Hessian or its approximation for many practical problems in engineering is not positive semidefinite, so the classical methods fail. Fortunately, a class of new methods (see [1–3]) using particular directions of negative curvature successfully overcomes the drawback. The key point of this idea is to produce a sequence $\{x_k\}$ whose limit point satisfies the second-order optimality conditions. This class of algorithms is called the second-order line search method.

The traditional second-order line search methods require the monotone descent of the objective values to guarantee their global convergence (see [4–6]). However, some researches [7–9] indicate that the monotone line search technique

☆ This work was supported by Project No. CityU 113308 of the Government of Hong Kong SAR, China; the National Natural Science Foundation of China No.10871098; the Natural Science Foundation of Jiangsu Province at grant BK2009397.

* Corresponding author.
E-mail addresses: hanxue726@163.com (X. Han), wysun@njnu.edu.cn, wysun2008@hotmail.com (W. Sun), mecdang@cityu.edu.hk (C. Dang).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.08.002

may have some drawbacks. For example, enforcing monotonicity may reduce the rate of convergence, especially when the iteration is near a narrow curved valley. Therefore, we allow the objective value with nonmonotone property in the iterative procedure. Grippo et al. [10] first proposed the nonmonotone line search for Newton's method. Due to the high efficiency of nonmonotonic methods, this technique has been utilized in many line search methods [11–13] and trust region methods [2,14–16]. Recently, the nonmonotone technique has been combined with the curvilinear search and second-order line search methods in [3,17,18]. The numerical results suggest that the new technique is efficient for unconstrained optimization problems.

The most popular line search techniques are the Armijo's rule, the Goldstein's rule and the Wolfe's rule. Unlike the Armijo's rule, the Goldstein's rule permits the increase of the stepsize occasionally. Furthermore, the Goldstein's rule can guarantee the objective function decrease sufficiently, while prevent the step length from being too small. However, the Goldstein's rule has a disadvantage that it is possible to exclude the minimizer value of the optimization problems as that pointed out in [19]. Therefore, the main aim of our work is to propose the nonmonotone second-order Wolfe's line search to overcome the drawbacks mentioned above.

The remainder of this paper is organized as follows. In Section 2, we describe the nonmonotone second-order Wolfe's line search and the new algorithm. In Section 3, we establish the convergence of the algorithm for unconstrained optimization. The numerical implementation is tested on a large set of standard test problems in Section 4.

2. The nonmonotone second-order Wolfe's rule

In this section, we will describe the nonmonotone second-order Wolfe's line search rule. First, we give the following definitions which can be found in [20].

Definition 2.1. A point x is called an indefinite point if $H(x)$ has at least one negative eigenvalue. Further, if x is an indefinite point, then d is a direction of the negative curvature if $d^T H(x)d < 0$.

Definition 2.2. Suppose s and d are nonascent directions and meanwhile d is a direction of the negative curvature, i.e., $s^T g(x) \le 0$, $d^T g(x) \le 0$, $d^T H(x)d < 0$, then (s, d) is called a descent pair at the indefinite point x; if x is not an indefinite point, then the pair (s, d) is called a descent pair if they satisfy $s^T g(x) \le 0$, $d^T g(x) \le 0$, $d^T H(x)d = 0$.

Definition 2.3. Let $\{\|s_k\|\}$ and $\{\|d_k\|\}$ be bounded, if

$$g_k^T s_k = 0 \text{ implies } g_k = 0 \text{ and } s_k = 0,$$
$$g_k^T s_k \to 0 \text{ implies } g_k \to 0 \text{ and } s_k \to 0,$$
$$d_k^T H(x)d_k \to 0 \text{ implies } \lambda_k \to 0 \text{ and } d_k \to 0,$$

where $\lambda_k = \min[0, \lambda_{\min}(H_k)]$, then $\{(s_k, d_k)\}$ is an acceptable sequence of the descent pair.

First, we describe the Wolfe's line search rule. When the step size α_k satisfies the following conditions:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho\alpha_k g_k^T d_k, \tag{2.1}$$
$$g_{k+1}^T d_k \ge \delta g_k^T d_k \tag{2.2}$$

where $\rho, \delta (0 < \rho < \delta < 1)$ are preassigned constants and d_k is the descent direction at x_k, then set $x_{k+1} = x_k + \alpha_k d_k$. Next, we recall the nonmonotone Wolfe's rule. Let

$$f(x_{l(k)}) = \max_{0 \le j \le m(k)} f(x_{k-j}),$$

where $m(k)$ satisfies $m(0) = 0$ and $0 \le m(k) \le \min\{m(k-1)+1, M\}$ for $k \ge 1$ and a nonnegative integer M. The traditional nonmonotone Wolfe's rule corresponding to (2.1)–(2.2) can be described as follows (see [8]):

$$f(x_k + \alpha_k d_k) \le \max_{0 \le j \le m(k)} \left[f(x_{k-j})\right] + \rho\alpha_k g_k^T d_k,$$
$$g(x_k + \alpha_k d_k)^T d_k \ge \delta g_k^T d_k,$$

where $0 < \rho < \delta < 1$. When x_k is an indefinite point, we can present the second-order Wolfe's rule. Let

$$x_k(\alpha_k) = x_k + \alpha_k^2 s_k + \alpha_k d_k,$$

where (s_k, d_k) is a descent pair at x_k. Replacing the Wolfe's rule (2.1)–(2.2), we require α_k to satisfy

$$f\left(x_k(\alpha_k)\right) \le f(x_k) + \rho\alpha_k^2 \left[s_k^T g_k + \frac{1}{2} d_k^T H_k d_k\right], \tag{2.3}$$
$$g\left(x_k(\alpha_k)\right)^T x_k'(\alpha_k) \ge \delta \left[g_k^T d_k + 2\alpha_k g_k^T s_k + \alpha_k d_k^T H_k d_k\right], \tag{2.4}$$

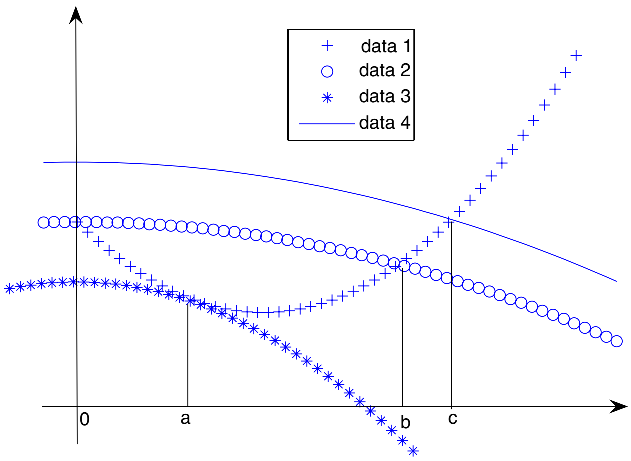

Fig. 1. Geometrical interpretation of (2.3)–(2.4) and (2.5)–(2.6).

where $0 < \rho < \delta < 1$. Corresponding to the second-order rule, we give the nonmonotone second-order rule:

$$f(x_k(\alpha_k)) \le f(x_{l(k)}) + \rho\alpha_k^2 \left[s_k^T g_k + \frac{1}{2} d_k^T H_k d_k \right], \tag{2.5}$$

$$g(x_k(\alpha_k))^T x_k'(\alpha_k) \ge \delta \left[g_k^T d_k + 2\alpha_k g_k^T s_k + \alpha_k d_k^T H_k d_k \right]. \tag{2.6}$$

Consequently, we only need to find an α_k satisfying (2.5)–(2.6), then set $x_k(\alpha_k) = x_k + \alpha_k^2 s_k + \alpha_k d_k$. When $M = 0$, (2.5)–(2.6) reduce to (2.3)–(2.4).

The nonmonotone second-order Wolfe's rule has a geometric interpretation as shown in Fig. 1. In Fig. 1, data 1 denote $f(x_k + \alpha^2 s_k + \alpha d_k)$, data 2 denote $f(x_k) + \rho\alpha^2[g_k^T s_k + \frac{1}{2} d_k^T H_k d_k]$, data 3 denote $c + \delta[\alpha g_k^T d_k + \alpha^2(g_k^T s_k + \frac{1}{2} d_k^T H_k d_k)]$, data 4 denote $f(x_{l(k)}) + \rho\alpha^2[g_k^T s_k + \frac{1}{2} d_k^T H_k d_k]$. We observe that (2.3) and (2.5) provide a sufficient decrease of the function. The rules (2.4) and (2.6) include the minimizing value of f. In addition, by use of (2.5) and (2.6), the obtained acceptable interval $[a, c]$ is larger than $[a, b]$, which satisfies (2.3)–(2.4).

Let

$$\phi_k(\alpha) = f\left(x_k + \alpha^2 s_k + \alpha d_k\right), \tag{2.7}$$

where (s_k, d_k) is a descent pair at x_k; then, $\phi_k'(0) = g_k^T d_k$, $\phi_k''(0) = 2g_k^T s_k + d_k^T H_k d_k$. The nonmonotone second-order Wolfe's rule (2.5)–(2.6) is equivalent to

$$\phi_k(\alpha_k) \le \phi_{l(k)}(0) + \frac{1}{2}\rho\alpha_k^2 \phi_k''(0) \tag{2.8}$$

and

$$\phi_k'(\alpha_k) \ge \delta\left[\phi_k'(0) + \alpha_k \phi_k''(0)\right]. \tag{2.9}$$

From Definition 2.3, we can obtain that $\phi_k'(0) < 0$ or $\phi_k'(0) \le 0$ and $\phi_k''(0) < 0$. The following lemma proves that such an α_k satisfying (2.5)–(2.6) exists under the conditions

$$s_k^T g_k < 0, \quad \text{whenever } g_k \ne 0, \tag{2.10}$$

and

$$d_k^T H d_k < 0, \quad \text{whenever } g_k = 0. \tag{2.11}$$

Once the descent pair (s_k, d_k) is impossible to be found, then the iteration will be terminated.

Lemma 2.4. *Let $\phi_k : R \to R$ be twice continuously differentiable, and suppose that $L = \{\alpha \in [0, +\infty) : \phi_k(\alpha) \le \phi_k(0)\}$ is compact. If $\phi_k'(0) < 0$, or if $\phi_k'(0) \le 0$ and $\phi_k''(0) < 0$, then there is an interval of step size satisfying*

$$\phi_k(\alpha_k) \le \phi_{l(k)}(0) + \frac{1}{2}\rho\alpha_k^2 \phi_k''(0), \tag{2.12}$$

$$\phi_k'(\alpha_k) \ge \delta\left[\phi_k'(0) + \alpha_k \phi_k''(0)\right]. \tag{2.13}$$

Proof. Let $\beta = \sup\{\alpha \in [0, +\infty) : \phi_k(\alpha) \le \phi_k(0)\}$. Since either $\phi_k'(0) < 0$ or $\phi_k'(0) \le 0$ and $\phi_k''(0) < 0$, then $\beta > 0$. Moreover, the compactness assumption of L and the continuity of ϕ_k imply that β is finite and $\phi_k(0) = \phi_k(\beta)$. Define h_1 and

$h_2 : R \to R$ by

$$h_1(\alpha) = \phi_k(\alpha) - \phi_k(0) - \frac{1}{2}\rho\alpha^2\phi_k''(0),$$
$$h_2(\alpha) = \phi_k'(\alpha) - \delta\left[\phi_k'(0) + \alpha\phi_k''(0)\right],$$
$$h_3(\alpha) = \phi_k(\alpha) - \phi_{l(k)}(0) - \frac{1}{2}\rho\alpha^2\phi_k''(0).$$

Obviously, h_1 and h_2 are also twice continuously differentiable. Note that

$$h_1(0) = \phi_k(0) - \phi_k(0) = 0,$$
$$h_1'(0) = \phi_k'(0) < 0$$

and

$$h_1(\beta) = \phi_k(\beta) - \phi_k(0) - \frac{1}{2}\rho\beta^2\phi_k''(0) = -\frac{1}{2}\rho\beta^2\phi_k''(0) > 0.$$

Together with the continuity of h_1, it implies that the existence of $\beta_1 \in (0, \beta]$, such that

$$h_1(\beta_1) = 0, \qquad h_1'(\beta_1) > 0$$

and

$$h_1(\alpha) < 0, \quad \forall\alpha \in (0, \beta_1). \tag{2.14}$$

On the other hand,

$$h_2(0) = \phi_k'(0) - \delta\phi_k'(0) = (1-\delta)\phi_k'(0) < 0,$$
$$\begin{aligned} h_2(\alpha) &= \phi_k'(\alpha) - \alpha\rho\phi_k''(0) + \alpha\rho\phi_k''(0) - \alpha\delta\phi_k''(0) - \delta\phi_k'(0) \\ &= h_1'(\alpha) + \alpha(\rho-\delta)\phi_k''(0) - \delta\phi_k'(0); \end{aligned}$$

then

$$h_2(\beta_1) = h_1'(\beta_1) + \beta_1(\rho-\delta)\phi_k''(0) - \delta\phi_k'(0) > 0.$$

Also,

$$h_2'(\alpha) = \phi_k''(\alpha) - \delta\phi_k''(0),$$

that is

$$h_2'(0) = \phi_k''(0) - \delta\phi_k''(0) = (1-\delta)\phi_k''(0) < 0.$$

By the continuity property of h_2, there is $\beta_2 > 0$, such that

$$h_2(\alpha) \geq 0, \quad \forall\alpha \in (\beta_2, \beta_1). \tag{2.15}$$

Since $\phi_k(0) \leq \phi_{l(k)}(0)$, by use of (2.14) and (2.15), we can obtain that $h_3(\alpha) < h_1(\alpha) < 0$, (2.8) and (2.9) hold when $\alpha \in (\beta_2, \beta_1)$. $\square$

Now, we state the new method with the nonmonotone second-order Wolfe's rule.

Algorithm 2.5. Step 0. Data: x_0, integer $M \geq 0$, $0 < \rho < \delta < 1$, $\alpha_{max} > 0$.
Step 1. Set $k = 0$, $m(0) = 0$, compute $f(x_0)$.
Step 2. Compute g_k and H_k. If the termination condition holds, stop.
Step 3. Compute the descent pair (s_k, d_k) and $f(x_{l(k)})$.
Step 4. Compute a step size $\alpha_k < \alpha_{max}$, such that the following conditions hold:

$$f(x_k(\alpha_k)) \leq f(x_l(k)) + \rho\alpha_k^2\left[s_k^T g_k + \frac{1}{2}d_k^T H_k d_k\right],$$
$$g(x_k(\alpha_k))^T x_k'(\alpha_k) \geq \delta\left[g_k^T d_k + 2\alpha_k g_k^T s_k + \alpha_k d_k^T H_k d_k\right].$$

Step 5. Set $x_{k+1} = x_k + \alpha_k^2 s_k + \alpha_k d_k$, and compute $f(x_{k+1})$, then set $m(k+1) = \min\{m(k)+1, M\}$, $k := k+1$, go to Step 2.

3. Convergence analysis

In this section, we establish the global convergence of Algorithm 2.5. We first make the following assumptions.

Assumption 3.1. (1) The level set $\Omega = \{x | f(x) \leq f(x_0)\}$ is compact and $f(x) : R^n \to R$ is a twice continuously differentiable function in Ω for any given $x_0 \in R^n$.
(2) The sequence $\{(s_k, d_k)\}$ in Algorithm 2.5 is acceptable.

Lemma 3.2. *Suppose that Assumption 3.1 holds and that $\{x_k\}$ is generated by Algorithm 2.5, then the sequence $\{x_k\}$ remains in Ω, and $\{f(x_{l(k)})\}$ is non-increasing and convergent.*

Proof. The proof is similar to that of Lemma 3.2 in [19]. □

Lemma 3.3. *Suppose that Assumption 3.1 holds, and that Algorithm 2.5 produces an infinite sequence $\{x_k\}$, then*

$$\lim_{k\to\infty} f(x_k) = \lim_{k\to\infty} f(x_{l(k)}), \tag{3.1}$$

$$\lim_{k\to\infty} \alpha_k^2\|s_k\| = 0, \qquad \lim_{k\to\infty} \alpha_k\|d_k\| = 0, \tag{3.2}$$

and

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{3.3}$$

Proof. The proof is similar to that of Lemma 3.3 in [19]. □

Theorem 3.4. *Suppose that Assumption 3.1 holds, then Algorithm 2.5 either terminates at some x_k such that*

$$g_k^T s_k = 0, \qquad d_k^T H_k d_k = 0, \tag{3.4}$$

or produces an infinite sequence such that

$$\lim_{k\to\infty} g_k^T s_k = 0, \qquad \lim_{k\to\infty} d_k^T H_k d_k = 0. \tag{3.5}$$

Proof. First, assume that the algorithm terminates in finitely many iterations, then for sufficiently large k, $g_k^T s_k = 0$, $d_k^T H_k d_k = 0$. On the other hand, we consider the case in which the sequence $\{x_k\}$ generated by Algorithm 2.5 is an infinite sequence.

Let

$$\phi_k(\alpha) = f\left(x_k + \alpha^2 s_k + \alpha d_k\right).$$

As the above stated, we have

$$\phi_k'(0) = g_k^T d_k \le 0$$

and

$$\phi_k''(0) = 2g_k^T s_k + d_k^T H_k d_k < 0.$$

Using Lemma 3.2, we know that the sequence $\{x_k\}$ remains in Ω. From Lemma 3.3 and (3.1), we have $\{f_{k+1} - f_{l(k)}\}$ converges to zero. It follows that

$$\lim_{k\to\infty} \alpha_k^2 g_k^T s_k = 0 \tag{3.6}$$

and

$$\lim_{k\to\infty} \frac{1}{2}\alpha_k^2 d_k^T H_k d_k = 0. \tag{3.7}$$

By use of the equality $\phi_k'(\alpha_k) \ge \delta[\phi_k'(0) + \phi_k''(0)\alpha_k]$, we have that

$$\phi_k'(\alpha_k) - \phi_k'(0) - \alpha_k\phi_k''(0) \ge -(1-\delta)\left[\phi_k'(0) + \alpha_k\phi_k''(0)\right]$$

and hence that

$$\phi_k'(\alpha_k) - \phi_k'(0) - \alpha_k\phi_k''(0) \ge -(1-\delta)\alpha_k\phi_k''(0).$$

Using the mean-value theorem yields that for some $\theta_k \in (0, \alpha_k)$,

$$\phi_k''(\theta_k) - \phi_k''(0) \ge -(1-\delta)\phi_k''(0). \tag{3.8}$$

Now, assume that at least one of $\lim_{k\to\infty} g_k^T s_k = 0$ and $\lim_{k\to\infty} d_k^T H_k d_k = 0$ does not hold. First we assume that $\lim_{k\to\infty} g_k^T s_k = 0$ does not hold, while $\lim_{k\to\infty} d_k^T H_k d_k = 0$ holds. Then, there is a subsequence $\{x_{k_i}\} \subset \{x_k\}$ and a constant ϵ, such that $g_{k_i}^T s_{k_i} \ge \epsilon$ and $-d_{k_i}^T H_{k_i} d_{k_i} < \epsilon$, which means

$$\left(g_{k_i}^T s_{k_i} + \frac{1}{2} d_{k_i}^T H_{k_i} d_{k_i}\right) > \frac{1}{2}\epsilon > 0,$$

Table 1
Test function.

P	Function name	P	Function name
1	Helical valley	11	Biggs exp 6
2	Beale	12	Penalty I
3	Brown and Dennis	13	Penalty II
4	Extended Rosenbrock	14	Cube
5	Extended Powell singular	15	Trigonometric
6	Gulf res. and dev	16	Variably dimensioned
7	Gaussian	17	Watson
8	Box 3-dimesional	18	Wood
9	Sc. Rosenbrock	19	Sc. Cube
10	Bro. Bad. Sc.	20	Chebyquad

that is to say,

$$\phi''_{k_i}(0) > \frac{1}{2}\epsilon > 0,$$

which contradicts $\phi''_k(0) < 0$.

Similarly, if $\lim_{k\to\infty} g_k^T s_k = 0$ holds, while $\lim_{k\to\infty} d_k^T H_k d_k = 0$ does not hold, we can obtain a contradiction similarly. Finally, we prove the contradiction if both of them do not hold. There is a subsequence $\{x_{k_i}\} \subset \{x_k\}$ and a constant ϵ, such that $g_{k_i}^T s_{k_i} \le -\epsilon$ and $d_{k_i}^T H_{k_i} d_{k_i} \le -\epsilon$, which means

$$-\left(g_{k_i}^T s_{k_i} + \frac{1}{2} d_{k_i}^T H_{k_i} d_{k_i}\right) \ge \frac{3}{2}\epsilon > 0,$$

that is to say,

$$-\phi''_{k_i}(0) > \frac{3}{2}\epsilon > 0. \tag{3.9}$$

Hence (3.8) implies that α_{k_i} does not converge to zero. In fact, if $\alpha_{k_i} \to 0$, then $\theta_{k_i} \to 0\,(k_i \to \infty)$, and $\phi''_{k_i}(\theta_{k_i}) - \phi''_{k_i}(0) \to 0$. We have $-(1-\delta)\phi''_k(0) \le 0$, which contradicts (3.9). On the other hand, if $\alpha_{k_i} \not\to 0\,(k_i \to \infty)$, under the assumption that $\lim_{k\to\infty} g_k^T s_k = 0$ and $\lim_{k\to\infty} d_k^T H_k d_k = 0$ do not hold, we know that (3.6) and (3.7) do not converge to zero, which gives a contradiction. The contradiction shows that the theorem is true. □

Remark 3.5. From Theorem 3.4, we obtain that either for a finite k, $g_k = 0$ and $\lambda_k = 0$, or $g_k \to 0$ and $\lambda_k \to 0$ for any sequence generated by Algorithm 2.5.

Lemma 3.6. *Let $f: R^n \to R$ be continuously differentiable on a compact set Ω. Assume that $f(x)$ has a finite number of critical points in Ω. Then, the sequence $\{x_k\} \subset \Omega$ satisfies*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0, \qquad \lim_{k\to\infty} \|g_k\| = 0, \tag{3.10}$$

and

$$\lim_{k\to\infty} x_k = x^*, \quad \text{and} \quad g(x^*) = 0. \tag{3.11}$$

Finally, we obtain directly that the sequence $\{x_k\}$ generated by Algorithm 2.5 converges to a second-order stationary point.

Theorem 3.7. *Suppose that Assumption 3.1 holds and that $f(x)$ has finitely many critical points in Ω. If $\{x_k\}$ generated by Algorithm 2.5 is an infinite sequence, then $\{x_k\}$ converges to some $x^* \in \Omega$ with $g(x^*) = 0$ and $\lambda_{min}(H(x^*)) \ge 0$.*

Since the proof is similar to Theorem 3.7 in [19], we omit it.

4. Numerical result

In this section, we evaluate the behavior of Algorithm 2.5 by using the Bunch–Parlett decomposition. We test on a set of standard test problems which appeared in [21]. In Table 1, we list the test functions. A MATLAB program is coded to perform the experiments.

We set the parameters $\rho_1 = 0.1$, $\delta = 0.2$, $\alpha_{max} = 10$. The stopping criterion is $\|g_k\| \le 10^{-5}$. The numerical results of our experiment are reported in Table 2. In Table 2 "NF-NG-NI" and "FVAL" denote the number of function evaluation, the number of gradient evaluation, the number of indefinite H_k appearing in the iterations of the algorithm, and the final objective function value, respectively. We denote the size of problems by N. The starting point is $10^L x_0$, where x_0 is the standard starting point. The sign '-' means that NF is more then 1000, and in this case we think that the algorithm fails (Table 2).

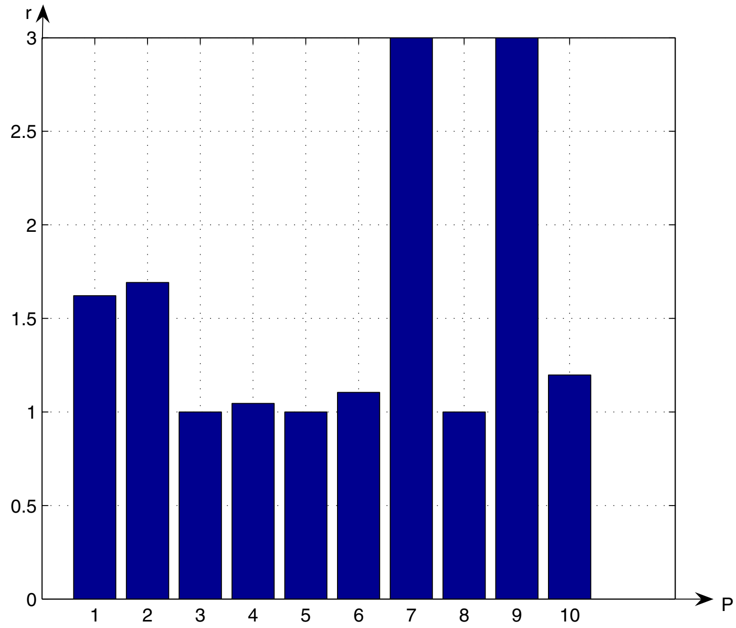

Fig. 2. Ratio of the iteration number of two methods for problem 1–10.

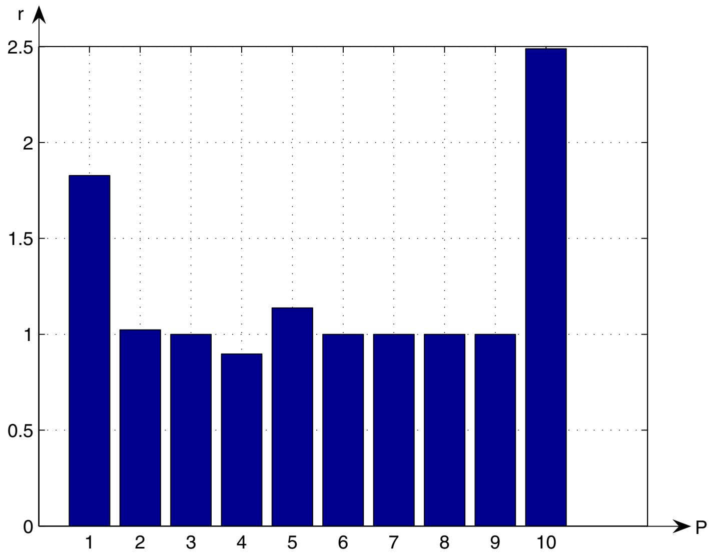

Fig. 3. Ratio of the iteration number of two methods for problem 11–20.

In Table 2, we compare the numerical results of monotone and nonmonotone algorithms. We run the program of Algorithm 2.5 with $M = 0$ (monotone second-order Wolfe's line search) and $M = 6$ (nonmonotone second-order Wolfe's line search) respectively. The result is listed in Table 2. In Figs. 2 and 3, we compare the nonmonotone second-order Wolfe's line search method (2.8)–(2.9) with the method of Goldstein's rule in [19], which is characterized by

$$f(x_k(\alpha_k)) \leq f(x_{l_{(k)}}) + \rho_1 \alpha_k^2 \left[s_k^T g_k + \frac{1}{2} d_k^T H_k d_k \right]$$

and

$$f(x_k(\alpha_k)) \geq f(x_k) + \rho_2 \alpha_k^2 \left[s_k^T g_k + \frac{1}{2} d_k^T H_k d_k \right],$$

where $\rho_1 = 0.1$, $\rho_2 = 1 - \rho_1$. Comparative experiments are performed on the same standard test functions and the same standard starting points.

Table 2
Numerical results for the new algorithm.

			M=0		M=6	
P	N	L	NF-NG-NI	FVAL	NF-NG-NI	FVAL
1	3	0	27-14-3	1.9349e−023	36-23-7	8.4491e−019
2	2	0	15-10-3	1.9514e−023	13-10-3	2.1580e−016
3	4	0	9-9-0	8.5822e+004	9-9-0	8.5822e+004
	4	1	15-15-0	8.5822e+004	15-15-0	8.5822e+004
	4	2	24-22-0	8.5822e+004	21-21-0	8.5822e+004
4	10	0	26-18-0	2.1677e−013	18-14-0	7.6782e−020
	10	1	107-52-0	8.4973e−013	9-8-0	1.0309e−022
	10	2	740-367-0	3.4310e−017	9-8-0	8.0858e−030
5	4	0	16-16-0	4.3788e−009	16-16-0	4.3788e−009
	4	1	22-22-0	2.6011e−009	22-22-0	2.6011e−009
	4	2	27-27-0	7.8223e−009	27-27-0	7.8223e−009
6	3	0	27-21-5	4.9860e−021	24-19-5	9.0252e−015
7	3	0	2-2-0	1.1293e−008	2-2-0	1.1293e−008
	3	1	56-18-12	1.1280e−008	22-8-7	1.1280e−008
8	3	0	13-11-2	9.9279e−011	13-11-2	9.9279e−011
9 ($c = 10^4$)	2	0	183-97-0	1.8674e−016	43-31-0	1.9044e−023
	2	1	742-358-0	4.9745e−019	9-8-0	1.2337e−028
	2	2	–	–	9-8-0	0
9 ($c = 10^6$)	2	0	–	–	43-31-0	1.9056e−023
	2	1	–	–	9-8-0	1.2326e−026
	2	2	–	–	9-8-0	0
10	2	1	42-21-13	1.9722e−031	42-21-13	1.9722e−031
11	6	0	71-36-35	0.0057	115-94-93	0.0057
12	4	0	44-23-0	2.2601e−005	17-17-0	2.2513e−005
	4	1	61-36-0	2.2500e−005	24-24-0	2.2518e−005
	10	0	81-29-0	7.0877e−005	22-21-0	7.0884e−005
13	10	0	402-207-0	8.7880e−006	19-19-0	8.8147e−006
14	2	0	60-22-0	2.0688e−014	31-13-0	1.8188e−027
	2	1	–	–	21-12-1	4.7509e−015
15	20	0	16-9-2	1.9791e−012	16-9-2	1.9791e−012
	60	0	71-20-10	1.4893e−007	48-2-7	1.4893e−007
	80	0	94-21-14	7.3761e−007	36-16-5	1.4696e−006
	100	0	100-24-13	1.8410e−006	48-20-6	6.9062e−007
16	10	1	17-17-0	1.6819e−014	17-17-0	1.6819e−014
	10	2	24-24-0	7.0625e−022	24-24-0	7.0625e−022
17	12	0	13-13-0	4.7224e−010	13-13-0	4.7224e−010
18	4	0	100-57-2	2.0164e−013	34-29-1	2.7319e−020
	4	1	58-43-1	6.4757e−019	43-36-2	5.0868e−016
	4	2	53-14-1	1.1963e−014	49-41-1	3.4274e−016
19 ($c = 10^6$)	4	0	–	–	31-13-0	5.5364e−021
	4	1	–	–	35-19-1	2.0444e−011
20	6	0	102-24-16	1.1842e−014	64-17-9	3.4527e−015

From Table 2, we see that, for all test problems, the new method is efficient and robust, especially for some ill-conditioned problems. For most of the test problems, it is obvious that the number of "NF-NG-NI" is reduced, or at least the number is the same as that of the monotone second-order Wolfe's line search method.

In Figs. 2 and 3, the x-axis corresponds to test problems 1–10 and 11–20 in Table 1, respectively. The y-axis is the ratio of the numbers of iteration with Goldstein's rule to Wolfe's rule:

$$r = \frac{(NF + NG + NI)_G}{(NF + NG + NI)_N}.$$

If the Goldstein's method fails, we set the ratio three, which is the maximum ratio. On the other hand, if the Wolfe's method fails, we set the ratio zero. We see that all bars surpass one except for the fourth problem in Fig. 3 (i.e., the 14th problem in Table 1), which indicates that using nonmonotone second-order Wolfe's method requires less numbers of iterations to converge than using nonmonotone second-order Goldstein's method for most problems. For example, for the 10th problem in Fig. 3 (i.e., the 20th problem in Table 1), the Goldstein's method takes roughly 2.5 times of the number of iteration of the Wolfe's method does.

5. Conclusions

In this paper, we propose a new method using the nonmonotone second-order Wolfe's rule for unconstrained optimization, which is an extension of the traditional nonmonotone Wolfe's rule and the curvilinear line search. We prove that the sequences generated by the new algorithm converge to a secondorder stationary point. Finally, we give detailed numerical experiments and numerical comparison to show that our algorithm is potentially efficient.

Acknowledgements

The authors would like to thank two anonymous referees and the editor-in-chief for their valuable suggestions and comments that improve greatly the quality of this manuscript.

References

[1] J.R. Bunch, B.N. Parlett, Direct methods for solving symmetic indefinite systems of linear equations, SIAM J. Numer. Anal. 8 (1971) 639–655.
[2] N.Y. Deng, Y. Xiao, F.I. Zhou, Nonmonotonic trust region algorithm, J. Optim. Theory Appl. 26 (1993) 259–285.
[3] M.C. Ferris, S. Lucidi, M. Roma, Nonmonotone curvilinear line search methods for unconstrained optimization, Comput. Optim. Appl. 6 (1996) 117–136.
[4] D. Goldfarb, Cuvilinear path steplength algorithms for minimization which use direcrion of negative curvature, Math. Program. 18 (1980) 31–40.
[5] J.J. Moré, D.C. Sorensen, On the use of direction of negative curvature in a modified Newton method, Math. Program. 16 (1979) 1–20.
[6] G.P. McCormick, A modification Armijo's step-size rule for negative curvature, Math. Program. 13 (1977) 111–115.
[7] L. Grippo, F. Lamparillo, S. Lucidi, A class of nonmonotone stabilization methods in unconstrained optimization, Numer. Math. 55 (1991) 779–805.
[8] W. Sun, J. Han, J. Sun, On global convergence of nonmonotone descent method, J. Comput. Appl. Math. 146 (2002) 89–98.
[9] H. Zhou, W. Sun, Nonmognotone descent algorithm for nonsmooth unconstrained optimization problems, Inter. J. Pure Appl. Math. 9 (2003) 153–163.
[10] L. Grippo, F. Lamparillo, S. Lucidi, A nonmonotone line search techinque for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.
[11] G. Liu, J. Han, D. Sun, Global convergence of BFGS algorithm with nonmonotone line search, Optimization 34 (1995) 147–159.
[12] Ph.L. Toint, An assessment of nonmonotone line search techinques for unconstrained optimization, SIAM J. Scientific Computing 17 (1996) 725–739.
[13] X. Yan, D. Han, W. Sun, A modified projection method with a new direction for solving variable inequalities, Appl. Math. Comput. 211 (2009) 118–129.
[14] W. Sun, Nonmonotone trust-region method for solving optimization problems, Appl. Math. Comput. 156 (2004) 159–174.
[15] J. Chen, W. Sun, R.J.B. Sampaio, Numerical research on sentivity of nonmonotone trust region algorithms to their parameters, Comput. & Math. Appl. 56 (2008) 2932–2940.
[16] J. Fu, W. Sun, Nonmonotone adaptive trust-region method for unconstrained optinization problems, Appl. Math. Comput. 163 (2005) 489–504.
[17] Y. Xiao, F.I. Zhou, Nonmonotone trust region method with curvilinear path in unconstrained minimization, Computing 48 (1992) 303–317.
[18] Q.Y. Zhou, W. Sun, Adaptive nonmonotone trust region method with curvilinear searches, J. Comput. Math. 24 (2006) 761–770.
[19] W. Sun, Q.Y. Zhou, An unconstrained optimization method using nonmonotone second order Goldstein's line search, Sci. China Ser. A: Math. 50 (2007) 1389–1400.
[20] W. Sun, Y. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer, New York, 2006.
[21] J.J. Moré, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software 7 (1981) 17–41.