Journal of Computational and Applied Mathematics 233 (2009) 519–530

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

A conjugate gradient method with descent direction for unconstrained optimization☆

Gonglin Yuan[a,*], Xiwen Lu[b], Zengxin Wei[a]

[a] College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, PR China

[b] School of Science, East China University of Science and Technology, Shanghai, 200237, PR China

ARTICLE INFO

Article history:
Received 3 December 2006
Received in revised form 13 March 2009

MSC:
65K05

Keywords:
Search direction
Line search
Conjugate gradient method
Global convergence
Unconstrained optimization

ABSTRACT

A modified conjugate gradient method is presented for solving unconstrained optimization problems, which possesses the following properties: (i) The sufficient descent property is satisfied without any line search; (ii) The search direction will be in a trust region automatically; (iii) The Zoutendijk condition holds for the Wolfe–Powell line search technique; (iv) This method inherits an important property of the well-known Polak–Ribière–Polyak (PRP) method: the tendency to turn towards the steepest descent direction if a small step is generated away from the solution, preventing a sequence of tiny steps from happening. The global convergence and the linearly convergent rate of the given method are established. Numerical results show that this method is interesting.

1. Introduction

The line search method often takes the following iterative formula

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots \tag{1.1}$$

for the unconstrained optimization problem

$$\min_{x \in \Re^n} f(x), \tag{1.2}$$

where $f : \Re^n \to \Re$ is continuously differentiable, x_k is the current iterate point, d_k is a search direction, $\alpha_k > 0$ is a steplength along d_k.

It is well known that there are many methods for solving optimization problems (see [1–9] etc), where the conjugate gradient method is a powerful line search method because of its simplicity and its very low memory requirement. This method can avoid, like steepest descent method, the computation and storage of some matrices associated with the Hessian of objective functions. The search direction d_k of the conjugate gradient method has the form

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

☆ This work is supported by China NSF grands 10761001 and the Scientific Research Foundation of Guangxi University (Grant No. X081082).

* Corresponding author. Tel.: +86 07713895307.
E-mail addresses: glyuan@tom.com, glyuan@gxu.edu.cn (G. Yuan).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.08.001

where $\beta_k \in \Re$ is a scalar which determines the different conjugate gradient methods, and g_k is the gradient of $f(x)$ at the point x_k. These methods in [10–15] are equivalent (see [16,17] etc) in the linear case, namely, when f is a strictly convex quadratic function and α_k is calculated by the following exact minimization rule: the step size α_k is chosen such that

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k). \tag{1.4}$$

At present, one of the most efficient formula for β_k is the following PRP method

$$\beta_k^{PRP} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|g_k\|^2}, \tag{1.5}$$

where g_k and g_{k+1} are the gradients $\nabla f(x_k)$ and $\nabla f(x_{k+1})$ of $f(x)$ at the point x_k and x_{k+1}, respectively, and $\|.\|$ denotes the Euclidian norm of vectors. Throughout this paper, we also denote $f(x_k)$ by f_k. Polak and Ribière [15] proved that the PRP conjugate gradient method with the exact line search is globally convergent when the objective function is convex. However, Powell [18] gave a counter example to show that there exist nonconvex functions on which the PRP method does not converge globally even the exact line search is used. He suggested that β_k should not be less than zero if we want to get the global convergence of the PRP method. Considering this suggestion, Gilbert and Nocedal [19] proved that the modified PRP method $\beta_k^+ = \max\{0, \beta_k^{PRP}\}$ is globally convergent under the sufficiently descent condition.

From the literature, one hopes to find the steplength α_k using the following weak Wolfe–Powell (WWP) line search

$$f(x_k + \alpha_k d_k) \leq f_k + \delta \alpha_k g_k^T d_k \tag{1.6}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{1.7}$$

where $\delta \in (0, 1/2), \sigma \in (\delta, 1)$. However, the global convergence of the PRP conjugate gradient method is still open with the above WWP conditions. Some formulas which possess the global convergence property (such as β_k^{DY} [10]) with the WWP did not perform better than the performances of the PRP method in numerical computation. Based β_k^{DY} and used the WWP conditions, Dai and Yuan [20] proposed an efficient conjugate gradient method. Over the past few years, much effort has been put to find out new formulas for conjugate methods such that they have not only global convergence for general functions but also good numerical performances (see [16,19]). Thus, any new conjugate gradient method should at least satisfy one of the following conditions [2]:

(i) The method with the WWP line search rule (or other line search rules) has some strongly convergent properties, at least, the method with the formula and the WWP line search rule (or other line search rules) may generate a descent direction at each iteration, and converges globally.

(ii) The average performances on the numerical computation of the formula with WWP line search rule (or others) should not be much inferior to the ones of the PRP.

In this paper, we design a new CG formula to satisfy the conditions above.

The following sufficiently descent condition

$$g_k^T d_k \leq -c\|g_k\|^2, \quad \forall\, k \geq 0 \text{ and some constant } c > 0 \tag{1.8}$$

is often used to analyze the global convergence of the nonlinear conjugate gradient method with the inexact line search techniques. For instance, Toouati-Ahmed and Storey [21], Al-Baali [22], Gilbert and Nocedal [19], Hu and Storey [23], and Gilbert and Nocedal [19] hinted that the sufficient descent condition may be crucial for conjugate gradient methods. In order to ensure the sufficiently descent condition and establish the convergence of the PRP method, Grippo and Lucidi [24] presented a new line search rule. There are some modified conjugate gradient methods which satisfied the sufficient descent conditions (see [25–29] etc). But for some methods which have been studied in the optimization area, such as the steepest descent method and the Newton method, the descent properties or the sufficient descent properties are independent of line searches. Is there any nonlinear conjugate gradient formula which possesses the sufficient descent property (1.8) without any line search? In this paper, we propose a method to answer this question positively. Furthermore, the search direction of the presented method will be in a trust region automatically and the scalar $\beta_k \geq 0$ is true.

In Section 2, the proposed algorithm is stated. The properties and the convergent results of the new method are given in Section 3. Numerical results and one conclusion are presented in Section 4 and in Section 5, respectively.

2. Algorithm

Motivated by the observation of Section 1, we describe a new nonlinear conjugate gradient algorithm as follows and call it Algorithm 1.

Algorithm 1. Step 0: Choose an initial point $x_0 \in \Re^n, \varepsilon \in (0, 1), 0 < \delta < \frac{1}{2}, \delta < \sigma < 1, c_0 \in (1, 2), \mu_1 \in (\frac{2c_0}{c_0 - 1}, +\infty), \mu_2 \in (1, +\infty)$. Set $d_0 = -g_0 = -\nabla f(x_0), k := 0$.

Step 1: If $\|g_k\| \leq \varepsilon$, then stop; Otherwise go to the next step.

Step 2: Compute step size α_k by line search rule (1.6) and (1.7), let $x_{k+1} = x_k + \alpha_k d_k$.

Step 3: If $\|g_{k+1}\| \leq \varepsilon$, then stop; Otherwise calculate the scalar β_k^{PRP} by (1.5).

Step 4: If $|\beta_k^{PRP}| \leq \frac{\|g_{k+1}\|}{\mu_2 \max\{\|g_k\|, \|d_k\|\}}$, let $\beta_k = \beta_k^{PRP}$ and go to Step 6, otherwise go to Step 5.

Step 5: Define a new scalar by

$$\beta_k^{\text{new}} = \frac{g_{k+1}^T \left(g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k\right)}{\mu_1 \max\{\|g_k\|\|g_{k+1}\|, |s_k^T y_k|\}}, \tag{2.1}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$. Let $\beta_k = \beta_k^{\text{new}}$.

Step 6: Let $d_{k+1} = -g_{k+1} + \beta_k d_k$, go to Step 7.

Step 7: Set $k := k + 1$, and go to Step 2.

Remark. (i) We define the following two index sets by

$$I_{PRP} = \left\{k \big| |\beta_k^{PRP}| \leq \frac{\|g_{k+1}\|}{\mu_2 \max\{\|g_k\|, \|d_k\|\}}\right\}, \qquad I_{\text{new}} = \left\{k \big| |\beta_k^{PRP}| > \frac{\|g_{k+1}\|}{\mu_2 \max\{\|g_k\|, \|d_k\|\}}\right\}.$$

(ii) If $x_{k+1} \approx x_k$ for $k, k+1 \in I_{\text{new}}$, we have $\|g_{k+1}\| \approx \|g_k\|$ implying that $\beta_k^{\text{new}} \to 0$ from (2.1), which means that $\beta_k \to 0$ if a small step is generated for all $k \geq 0$. Thus the given method inherits the better property of the PRP method: the directions will turn out to be the steepest descent directions if the tiny steps from happening.

(iii) From Step 4 in Algorithm 1, $\beta_k = \beta_k^{\text{new}} = \frac{g_{k+1}^T (g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k)}{\mu_1 \max\{\|g_k\|\|g_{k+1}\|, |s_k^T y_k|\}}$ when $k \notin I_{PRP}$ holds, by the exact minimization rule (1.4), it is not difficult to deduce that the conjugacy condition $d_{k+1}^T \nabla^2 f(x_k) d_k = 0$ holds. Otherwise, $\beta_k = \beta_k^{PRP}$ at the kth iteration. Overall, this method cannot reduce to the linear conjugate gradient method, then we can only say that it is some kind of memory gradient method.

3. Properties and convergence analysis

This section will report some better properties and convergence of the new algorithm. In the following, we assume that $g_k \neq 0$ for all k, for otherwise a stationary point has been found.

It has been showed that the PRP method with the strong Wolfe–Powell (SWP) line search rules does not ensure the condition (1.8) at each iteration. The following lemma shows that the search direction d_k will satisfy condition (1.8) and be in a trust region without carrying out any line search technique.

Lemma 3.1. *Let the sequence $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1, then there exist constants $c > 0$ and $c_1 > 0$ such that (1.8) and*

$$\|d_k\| \leq c_1 \|g_k\| \tag{3.1}$$

for all $k \geq 0$.

Proof. If $k = 0$, this result is obtained obviously. If $k \geq 1$, we discuss the problem divided into the following two cases.

Case 1. If $k \in I_{PRP}$, then we have

$$|\beta_k| = |\beta_k^{PRP}| \leq \frac{\|g_{k+1}\|}{\mu_2 \max\{\|g_k\|, \|d_k\|\}} \leq \frac{\|g_{k+1}\|}{\mu_2 \|d_k\|}, \quad \mu_2 \in (1, +\infty).$$

This together with Step 6 of Algorithm 1, we get

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \beta_k d_k^T g_{k+1} \leq -\|g_{k+1}\|^2 + |\beta_k| \|d_k\| \|g_{k+1}\| \leq -\left(1 - \frac{1}{\mu_2}\right) \|g_{k+1}\|^2$$

and

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k| \|d_k\| \leq \|g_{k+1}\| + \frac{\|g_{k+1}\|}{\mu_2 \|d_k\|} \|d_k\| = \left(1 + \frac{1}{\mu_2}\right) \|g_{k+1}\|.$$

Case 2. If $k \in I_{\text{new}}$, by Step 5 of Algorithm 1, we get

$$|\beta_k| = |\beta_k^{\text{new}}| \leq \frac{\|g_{k+1}\|(\|g_{k+1}\| + \frac{\|g_{k+1}\|}{\|g_k\|}\|g_k\|)}{\mu_1 \max\{\|g_k\|\|g_{k+1}\|, |s_k^T y_k|\}} \leq \frac{\|g_{k+1}\|}{\mu_1 \|g_k\|}, \qquad \mu_1 \in \left(\frac{2c_0}{c_0 - 1}, +\infty\right), \qquad c_0 \in (1, 2).$$

In this case, we can show that $\|d_k\| \leq c_0\|g_k\|$ holds for all $k \in I_{\text{new}}$. Now we let $\|d_k\| \leq c_0\|g_k\|$, and prove that $\|d_{k+1}\| \leq c_0\|g_{k+1}\|$ is true too. Consider Step 6 of Algorithm 1, we obtain

$$\begin{aligned}\|d_{k+1}\| &\leq \|g_{k+1}\| + |\beta_k|\|d_k\| \leq \left(1 + \frac{c_0}{\mu_1}\right)\|g_{k+1}\| \\ &\leq \left(1 + \frac{c_0(c_0-1)}{2c_0}\right)\|g_{k+1}\| = \frac{(c_0+1)}{2}\|g_k\| \\ &\leq \frac{(c_0+c_0)}{2}\|g_k\| = c_0\|g_{k+1}\|.\end{aligned}$$

Combining this with Step 6 of Algorithm 1, we have

$$\begin{aligned}d_{k+1}^T g_{k+1} &= -\|g_{k+1}\|^2 + \beta_k d_k^T g_{k+1} \\ &\leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|}{\mu_1\|g_k\|}\|d_k\|\|g_{k+1}\| \\ &\leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|}{\mu_1\|g_k\|}c_0\|g_k\|\|g_{k+1}\| \\ &= -\left(1 - \frac{c_0}{\mu_1}\right)\|g_{k+1}\|^2.\end{aligned}$$

Let $c = \min\{1 - \frac{1}{\mu_2}, 1 - \frac{c_0}{\mu_1}\}$ and $c_1 = \max\{c_0, 1 + \frac{1}{\mu_2}\}$, then (1.8) and (3.1) hold for all $k \geq 0$. The proof is complete. □

(3.1) shows that d_k is in a trust region. Then the new method has both the advantage of simple structure of conjugate gradient method and the advantage of strong convergence of trust region methods in some sense.

In fact, we can obtain that $\beta_k \geq 0$ for all $k \in I_{\text{new}}$

$$\beta_k = \beta_k^{\text{new}} = \frac{g_{k+1}^T(g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|}g_k)}{\mu_1 \max\{\|g_k\|\|g_{k+1}\|, |s_k^T y_k|\}} \geq \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|^2}{\|g_k\|}\|g_k\|}{\mu_1 \max\{\|g_k\|\|g_{k+1}\|, |s_k^T y_k|\}} = 0.$$

The following assumptions are often needed to prove the global convergence of the conjugate gradient methods.

Assumption 3.1. (i) The function $f(x)$ has a lower bound on $\Re^n$.

(ii) In an open convex set Ω_0 that contains the level set $\Omega = \{x \in \Re^n \mid f(x) \leq f(x_0)\}$, where x_0 is given, f is differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \Omega_0. \tag{3.2}$$

Theorem 3.1. *Let $\{\alpha_k, d_k, x_{k+1}, g_{k+1}\}$ be generated by Algorithm 1 with the Wolfe–Powell line search rules (1.6) and (1.7), and f satisfies Assumption 3.1. Then the following Zoutendijk [30] condition*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \tag{3.3}$$

holds. Moreover we have

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.4}$$

Proof. By Assumption 3.1(ii) and (1.7), we have

$$-(1-\sigma)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq \|g_{k+1} - g_k\|\|d_k\| \leq \alpha_k L\|d_k\|^2. \tag{3.5}$$

By Assumption 3.1(i) and $d_k^T g_k < 0$, we can obtain that $\{f_k\}$ is a decreasing sequence and has a lower bound, thus $\{f_k\}$ is a convergent sequence. Using (1.6) and (3.5), we have

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty.$$

Moreover, by Lemma 3.1, we get (3.4). The proof of this theorem is complete. □

Lemma 3.2. *Assume that f is twice continuously differentiable and uniformly convex on $\Re^n$, then Assumption 3.1 holds, $f(x)$ has a unique minimal point x^*, and there exist constants $0 < m \leq M$ such that*

$$m\|y\|^2 \le y^T\nabla^2 f(x)y \le M\|y\|^2, \quad \forall x,\ y \in \Re^n, \tag{3.6}$$

$$\frac{m}{2}\|x-x^*\|^2 \le f(x)-f(x^*) \le \frac{M}{2}\|x-x^*\|^2, \quad \forall x \in \Re^n \tag{3.7}$$

and

$$m\|x-x^*\| \le \|g(x)\| \le M\|x-x^*\|, \quad \forall x \in \Re^n. \tag{3.8}$$

Proof. Omitted. For the proof, see (e.g., [31,32], etc.). □

Theorem 3.2. *Let f be twice continuously differentiable and uniformly convex on $\Re^n$. Then the sequence $\{x_k\}$ generated by Algorithm 1 converges to x^* at least linearly.*

Proof. By Lemma 3.1, there exists a constant $\rho_0 = \frac{c}{c_1} \in (0, 1)$ satisfying

$$-\frac{g_k^T d_k}{\|d_k\|\|g_k\|} \ge \rho_0. \tag{3.9}$$

Using Lemma 3.1, (3.9), (1.6), (1.7), (3.7) and (3.8), there exists a constant $\rho_1 = \frac{\delta(1-\sigma)}{L}$ such that

$$\begin{aligned} f_k - f_{k+1} &\ge \rho_1\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2 = \rho_1\left(\frac{-g_k^T d_k}{\|d_k\|\|g_k\|}\right)^2\|g_k\|^2 \\ &\ge \rho_1\rho_0^2\|g_k\|^2 \ge \rho_1\rho_0^2 m^2\|x_k - x^*\|^2 \\ &\ge \frac{2\rho_1\rho_0^2 m^2}{M}(f_k - f(x^*)). \end{aligned} \tag{3.10}$$

It follows from (3.2) that $m \le L$. Then we have

$$\eta = \frac{2\rho_1\rho_0^2 m^2}{M} = \frac{2\delta(1-\sigma)\rho_0^2 m^2}{LM} \le \frac{(1-\sigma)\rho_0^2 m}{M} \le (1-\sigma)\rho_0^2 < 1.$$

Which combining with (3.10), we get

$$f_k - f(x^*) \le (1-\eta)(f_{k-1} - f(x^*)) \le (1-\eta)^2(f_{k-2} - f(x^*)) \le \cdots \le (\sqrt{1-\eta})^{2k}(f_0 - f(x^*)),$$

therefore, by (3.7), set $\eta_1 = \sqrt{1-\eta}$ and $\eta_2 = \sqrt{\frac{2(f_0 - f(x^*))}{m}}$, it follows that

$$\|x_k - x^*\|^2 \le \frac{2}{m}(f_k - f(x^*)) \le \frac{2(f_0 - f(x^*))}{m}(\sqrt{1-\eta})^{2k} = \eta_1^{2k}\eta_2^2,$$

thus we obtain

$$\|x_k - x^*\| \le \eta_2\eta_1^k,$$

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

4. Numerical results

This section reports some numerical experiments with Algorithm 1, the FR conjugate gradient method and the PRP conjugate gradient method. We test these algorithms on some problems [33] taken from MATLAB with given initial points and Benchmark Problems, respectively. The parameters common to these methods were set identically, $c_0 = 1.9$, $\mu_1 = \frac{40c_0}{1-c_0}$, $\varepsilon = 10^{-5}$, $\mu_2 = \frac{1}{1-\varepsilon}$, $\delta = 0.1$, $\sigma = 0.9$.

The following *Himmeblau* stop rule is used:

If $|f(x_k)| > e_1$, let $stop1 = \frac{|f(x_k)-f(x_{k+1})|}{|f(x_k)|}$; Otherwise, let $stop1 = |f(x_k) - f(x_{k+1})|$.

If $\|g(x)\| < \varepsilon$ or $stop1 < e_2$ was satisfied, we will stop the program, where $e_1 = e_2 = 10^{-5}$. We also stop the program if the iteration number is more than one thousand.

The columns of Table 1 have the following meaning:

Problem: the name of the test problem [33];

Dim: the dimension of the problem; NG: the number of the gradient evaluations;

NF: the number of the function evaluations; NI: the total number of iterations;

Dolan and Moré [34] gave a new tool to analyze the efficiency of Algorithms. They introduced the notion of a performance profile as a means to evaluate and compare the performance of the set of solvers S on a test set P. Assuming that there exist n_s solvers and n_p problems, for each problem p and solver s, they defined

Table 1
The line search is used by Wolfe–Powell rule. Test results for Algorithm 1/FR/PRR.

Problems	Dim	Algorithm 1 NI/NF/NG	FR NI/NF/NG	PRP NI/NF/NG
ROSE	2	23/157/68	59/188/115	38/164/88
FROTH	2	22/55/34	14/36/24	3/4/3
BADSCP	2	4/21/5	3/6/4	4/21/5
BADSCB	2	13/32/21	3/19/6	3/19/6
BEALE	2	17/36/21	19/43/26	5/10/7
JENSAM	2	3/6/4	3/19/4	3/6/4
HELIX	3	63/154/72	509/1221/675	61/156/73
BARD	3	32/62/33	70/145/76	45/93/48
GAUSS	3	2/3/2	2/3/2	2/3/2
MEYER	3	4/35/5	3/11/3	8/61/10
GULF	3	2/2/2	2/2/2	2/2/2
BOX	3	2/17/3	2/17/3	2/17/3
SING	4	73/155/85	66/157/93	41/105/66
WOOD	4	23/60/38	17/70/28	20/63/39
KOWOSB	4	82/149/87	97/207/108	40/76/41
BD	4	19/53/36	27/77/52	17/47/32
OSB1	5	2/17/3	2/17/3	2/17/3
OSB2	11	3/19/4	3/19/4	3/19/4
WATSON	2	11/25/16	11/24/15	11/25/16
ROSEX	10	28/198/80	120/387/236	28/138/68
SINGX	16	45/104/61	97/233/138	68/156/91
	80	104/219/120	87/218/133	83/204/123
PEN1	2	10/21/12	8/19/11	10/21/12
	100	25/79/44	37/92/57	32/91/54
PEN2	50	7/30/13	23/66/43	7/30/13
	200	2/2/2	2/2/2	2/2/2
VARDIM	50	3/18/5	3/18/5	3/18/5
	200	3/18/5	3/18/5	3/18/5
TRIG	50	30/63/32	67/174/106	30/104/41
	200	14/31/16	39/111/71	15/33/17
BV	50	2/3/2	2/3/2	2/3/2
	200	2/3/2	2/3/2	2/3/2
IE	50	5/9/5	5/9/5	5/9/5
	200	6/10/6	6/10/6	8/15/9
TRID	10	14/31/19	22/50/30	16/39/22
	800	21/46/27	162/361/201	21/46/27
BAND	100	13/30/19	14/32/20	12/42/19
	500	16/60/42	13/31/20	12/29/19
	800	12/29/19	13/31/20	13/33/21
LIN	100	2/3/3	2/3/3	2/3/3
	800	2/3/3	2/3/3	2/3/3
LIN1	100	4/6/4	4/6/4	4/6/4
	500	3/5/4	3/5/4	3/6/5
	1000	3/5/4	3/5/4	3/9/4
LIN0	100	4/20/6	3/4/3	4/20/6
	500	3/5/4	3/5/4	3/19/15
	800	3/5/4	3/5/4	3/6/5

$t_{p,s}$ = computing time (the number of function evaluations or others) required to solve problem p by solver s.

Requiring a baseline for comparisons, they compared the performance on problem p by solver s with the best performance by any solver on this problem; that is, using the performance ratio

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

Suppose that a parameter $r_M \geq r_{p,s}$ for all p, s is chosen, and $r_{p,s} = r_M$ if and only if solver s does not solve problem p.

The performance of solver s on any given problem might be of interest, but we would like to obtain an overall assessment of the performance of the solver, then they defined

$$\rho_s(t) = \frac{1}{n_p}\text{size}\{p \in P : r_{p,s} \leq t\},$$

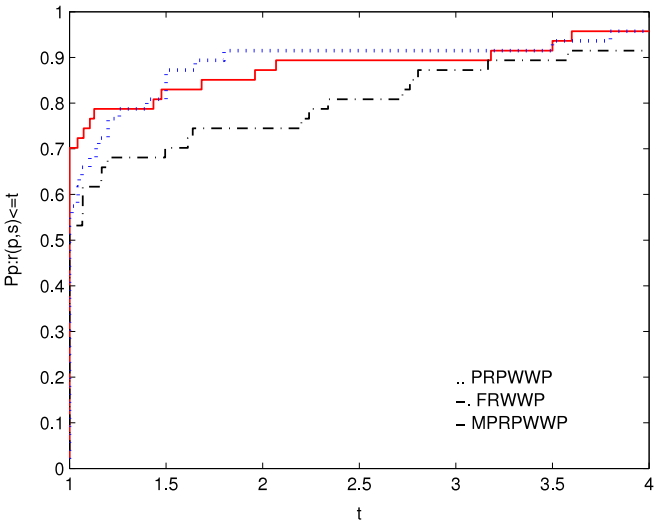

Fig. 1. Performance profiles (NF) of conjugate gradient methods with WWP.

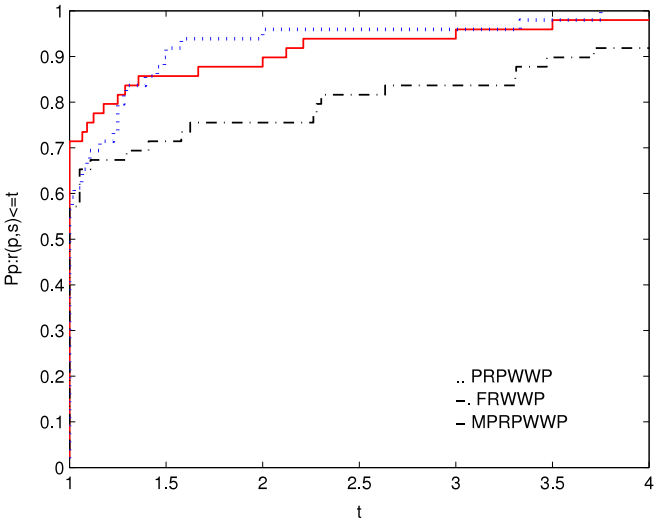

Fig. 2. Performance profiles (NG) of conjugate gradient methods with WWP.

thus $\rho_s(t)$ was the probability for solver $s \in S$ that a performance ratio $r_{p,s}$ was within a factor $t \in R$ of the best possible ration. Then function ρ_s was the (cumulative) distribution function for the performance ratio. The performance profile $\rho_s : R \mapsto [0, 1]$ for a solver was a nondecreasing, piecewise constant function, continuous from the right at each breakpoint. The value of $\rho_s(1)$ was the probability that the solver would win over the rest of the solvers.

According to the above rules, we know that one solver whose performance profile plot is on top right will win over the rest of the solvers.

Figs. 1 and 2 show that the performance of these methods is relative to *NF* and *NG* of Table 1, respectively.

Fig. 1 shows that the new method is better than PRP method for $t < 1.5$, and these two methods can solve about 96% of the test problems successfully. Fig. 2 shows that the PRP method is superior to the modified algorithm and the FR method, and the MPRP method is better than the FR method. From these two figures, it is clear that the FR method has the most fails (has the lowest probability of being the optimal solver).

Benchmark Problems: 1. Sphere function.

$$f_{Sph}(x) = \sum_{i=1}^{n} x_i^2, \qquad x_i \in [-5.12, 5.12]$$
$$x^* = (0, 0, \ldots, 0), \qquad f_{Sph}(x^*) = 0.$$

Table 2
Test results for FR method.

Sphere	x_0	NI/NFG/$f(\bar{x})$ $(-5,-5,\ldots,-5)$	NI/NFG/$f(\bar{x})$ $(-3,-3,\ldots,-3)$	NI/NFG/$f(\bar{x})$ $(-1,-1,\ldots,-1)$	NI/NFG/$f(\bar{x})$ $(2,2,\ldots,2)$	NI/NFG/$f(\bar{x})$ $(4,4,\ldots,4)$
Dim	10	2/6/7.888609e−030	3/7/7.888609e−030	3/7/1.109336e−030	3/7/4.930381e−031	2/6/1.972152e−030
	100	2/6/6.389773e−027	2/6/1.546167e−026	2/6/7.888609e−029	2/6/1.597443e−027	2/6/1.135960e−026
	300	2/6/6.399240e−025	2/6/6.058452e−026	2/6/3.328007e−027	2/6/5.916457e−027	2/6/1.599810e−025
Sphere	x_0	$(-5,0,-5,0,\ldots)$	$(-3,0,-3,0,\ldots)$	$(-1,0,-1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(4,0,4,0,\ldots)$
Dim	10	2/6/6.310887e−029	3/7/0.000000e+000	3/7/0.000000e+000	3/7/9.860761e−031	3/7/8.874685e−030
	100	2/6/3.944305e−027	2/6/9.860761e−028	3/7/6.162976e−031	2/6/2.465190e−028	2/6/1.577722e−028
	300	2/6/1.431783e−026	2/6/1.848893e−026	2/6/5.990412e−028	2/6/5.798128e−027	2/6/3.833864e−026
Schwefel's	x_0	$(-0.001,\ldots,-0.001)$	$(-0.0002,\ldots,-0.0002)$	$(0.0006,\ldots,0.0006)$	$(0.0007,\ldots,0.0007)$	$(0.0008,\ldots,0.0008)$
Dim	10	3/8/3.539977e−007	3/8/1.415991e−008	3/8/1.274392e−007	3/8/1.734589e−007	3/8/2.265585e−007
	50	5/14/2.297452e−006	4/11/3.471788e−007	5/14/8.270826e−007	5/14/1.125751e−006	5/14/1.470369e−006
	100	7/20/2.554559e−006	5/14/7.557030e−007	6/17/2.317420e−006	6/17/3.154267e−006	7/20/1.634918e−006
Schwefel's	x_0	$(-0.001,0,\ldots)$	$(-0.0002,0,\ldots)$	$(0.0006,0,\ldots)$	$(0.0007,0,\ldots)$	$(0.0008,0,\ldots)$
Dim	10	3/8/6.244149e−007	2/5/5.186065e−008	3/8/2.247894e−007	3/8/3.059633e−007	3/8/3.996256e−007
	50	5/14/3.464888e−006	3/8/6.064742e−007	4/11/1.848353e−006	4/11/2.515814e−006	4/11/3.285961e−006
	100	7/20/6.523845e−006	4/11/9.360044e−007	5/14/3.845945e−006	6/17/3.745873e−006	6/17/4.892569e−006
Rastrigin	x_0	$(-5,-5,\ldots,-5)$	$(-4,-4,\ldots,-4)$	$(1,1,\ldots,1)$	$(2,2,\ldots,2)$	$(3,3,\ldots,3)$
Dim	10	4/13/2.487372e+002	4/13/1.591924e+002	7/26/2.545001e−007	4/13/3.979831e+001	4/13/8.954601e+001
	100	3/9/2.487372e+003	3/9/1.591924e+003	3/9/9.949591e+001	3/9/3.979831e+002	3/9/8.954601e+002
	300	3/9/7.462117e+003	3/9/4.775773e+003	3/8/2.984877e+002	3/8/1.193950e+003	3/9/2.686380e+003
Rastrigin	x_0	$(-5,0,-5,0,\ldots)$	$(-4,0,-4,0,\ldots)$	$(1,0,1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(3,0,3,0,\ldots)$
Dim	10	3/9/1.243686e+002	3/9/7.959622e+001	3/9/4.974795e+000	3/9/1.989916e+001	3/9/4.477301e+001
	100	4/12/1.243686e+003	4/12/7.959622e+002	4/12/4.974795e+001	4/12/1.989916e+002	4/12/4.477301e+002
	300	3/9/3.731058e+003	3/9/2.387887e+003	3/9/1.492439e+002	3/9/5.969747e+002	3/9/1.343190e+003
Schwefel	x_0	$(-500,-500,\ldots,-500)$	$(-400,-400,\ldots,-400)$	$(100,100,\ldots,100)$	$(400,400,\ldots,400)$	$(500,500,\ldots,500)$
Dim	10	3/8/-9.940703e+003	2/6/-4.174987e+003	2/19/3.101787e+003	3/38/-1.569312e+003	2/19/5.780458e+002
	100	3/8/-9.771601e+004	2/6/-4.174488e+004	2/19/3.101787e+004	2/19/NaN	2/19/5.780458e+003
	300	3/8/-2.919275e+005	2/6/-1.252317e+005	2/19/9.305360e+004	3/37/5.534365e+005	2/19/1.734137e+004
Schwefel	x_0	$(-500,0,-500,0,\ldots)$	$(-400,0,-400,0,\ldots)$	$(100,0,100,0,\ldots)$	$(400,0,400,0,\ldots)$	$(500,0,500,0,\ldots)$
Dim	10	1/2/5.995721e+003	1/2/5.380480e+002	1/2/3.645808e+003	1/2/7.841610e+003	1/2/2.383937e+003
	100	1/2/5.995721e+004	1/2/5.380480e+003	1/2/3.645808e+004	1/2/7.841610e+004	1/2/2.383937e+004
	300	1/2/1.798716e+005	1/2/1.614144e+004	1/2/1.093742e+005	1/2/2.352483e+005	1/2/7.151812e+004
Griewank	x_0	$(-50,-50,\ldots,-50)$	$(-10,-10,\ldots,-10)$	$(1,1,\ldots,1)$	$(20,20,\ldots,20)$	$(30,30,\ldots,30)$
Dim	10	2/19/7.250909e+000	2/19/1.264953e+000	7/78/1.117345e+000	588/1956/9.801891e−006	3/9/2.543092e−006
	100	9/56/4.439395e−003	13/38/6.939268e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	36/108/7.106439e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
Griewank	x_0	$(-50,0,-50,0,\ldots)$	$(-10,0,-10,0,\ldots)$	$(1,0,1,0,\ldots)$	$(20,0,20,0,\ldots)$	$(30,0,30,0,\ldots)$
Dim	10	2/19/4.553192e+000	4/39/1.115204e+000	3/24/2.429039e−001	2/19/1.551065e+000	5/59/NaN
	100	5/35/2.672586e−004	59/177/7.300285e−006	3/8/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	41/123/8.360227e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000

2. Schwefel's function.

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

$$x^* = (0, 0, \ldots, 0), \qquad f_{SchDS}(x^*) = 0.$$

3. Rastrigin function.

$$f_{Ras}(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)), \qquad x_i \in [-5.12, 5.12]$$

$$x^* = (0, 0, \ldots, 0), \qquad f_{Ras}(x^*) = 0.$$

4. Schwefel function.

$$f_{Sch}(x) = 418.9829n + \sum_{i=1}^{n} x_i \sin\sqrt{|x_i|}, \qquad x_i \in [-512.03, 511.97]$$

$$x^* = (-420.9678, -420.9678, \ldots, -420.9678), \qquad f_{Sch}(x^*) = 0.$$

Table 3
Test results for PRP method.

Sphere	x_0	NI/NFG/$f(\bar{x})$ $(-5,-5,\ldots,-5)$	NI/NFG/$f(\bar{x})$ $(-3,-3,\ldots,-3)$	NI/NFG/$f(\bar{x})$ $(-1,-1,\ldots,-1)$	NI/NFG/$f(\bar{x})$ $(2,2,\ldots,2)$	NI/NFG/$f(\bar{x})$ $(4,4,\ldots,4)$
Dim	10	2/6/7.888609e−030	3/7/0.000000e+000	3/7/1.232595e−031	3/7/4.437343e−030	2/6/1.972152e−030
	100	2/6/6.389773e−027	2/6/1.546167e−026	2/6/7.888609e−029	2/6/1.597443e−027	2/6/1.135960e−026
	300	2/6/6.399240e−025	2/6/6.058452e−026	2/6/3.328007e−027	2/6/5.916457e−027	2/6/1.599810e−025
Sphere	x_0	$(-5,0,-5,0,\ldots)$	$(-3,0,-3,0,\ldots)$	$(-1,0,-1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(4,0,4,0,\ldots)$
Dim	10	2/6/6.310887e−029	3/7/9.860761e−031	3/7/6.162976e−032	3/7/0.000000e+000	3/7/9.860761e−031
	100	2/6/3.944305e−027	2/6/9.860761e−028	3/7/2.218671e−029	2/6/2.465190e−028	2/6/1.577722e−028
	300	2/6/1.431783e−026	2/6/1.848893e−026	2/6/5.990412e−028	2/6/5.798128e−027	2/6/3.833864e−026
Schwefel's	x_0	$(-0.001,\ldots,-0.001)$	$(-0.0002,\ldots,-0.0002)$	$(0.0006,\ldots,0.0006)$	$(0.0007,\ldots,0.0007)$	$(0.0008,\ldots,0.0008)$
Dim	10	3/8/3.539977e−007	3/8/1.415991e−008	3/8/1.274392e−007	3/8/1.734589e−007	3/8/2.265585e−007
	50	5/14/2.297452e−006	4/11/3.471788e−007	5/14/8.270826e−007	5/14/1.125751e−006	5/14/1.470369e−006
	100	7/20/2.554559e−006	5/14/7.557030e−007	6/17/2.317420e−006	6/17/3.154267e−006	7/20/1.634918e−006
Schwefel's	x_0	$(-0.001,0,\ldots)$	$(-0.0002,0,\ldots)$	$(0.0006,0,\ldots)$	$(0.0007,0,\ldots)$	$(0.0008,0,\ldots)$
Dim	10	3/8/6.244149e−007	2/5/5.186065e−008	3/8/2.247894e−007	3/8/3.059633e−007	3/8/3.996256e−007
	50	5/14/3.464888e−006	3/8/6.064742e−007	4/11/1.848353e−006	4/11/2.515814e−006	4/11/3.285961e−006
	100	7/20/6.523845e−006	4/11/9.360044e−007	5/14/3.845945e−006	6/17/3.745873e−006	6/17/4.892569e−006
Rastrigin	x_0	$(-5,-5,\ldots,-5)$	$(-4,-4,\ldots,-4)$	$(1,1,\ldots,1)$	$(2,2,\ldots,2)$	$(3,3,\ldots,3)$
Dim	10	3/23/2.487372e+002	3/23/1.591924e+002	4/14/9.949591e+000	3/12/3.979831e+001	4/15/8.954601e+001
	100	3/10/2.487372e+003	3/10/1.591924e+003	3/10/9.949591e+001	3/10/3.979831e+002	3/10/8.954601e+002
	300	3/10/7.462117e+003	3/10/4.775773e+003	3/10/2.984877e+002	3/10/1.193949e+003	3/10/2.686380e+003
Rastrigin	x_0	$(-5,0,-5,0,\ldots)$	$(-4,0,-4,0,\ldots)$	$(1,0,1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(3,0,3,0,\ldots)$
Dim	10	3/11/1.243686e+002	3/11/7.959622e+001	3/11/4.974795e+000	3/11/1.989916e+001	3/11/4.477301e+001
	100	3/23/1.243686e+003	3/10/7.959622e+002	3/10/4.974795e+001	3/10/1.989916e+002	3/10/4.477301e+002
	300	3/10/3.731058e+003	3/10/2.387887e+003	3/10/1.492439e+002	3/10/5.969747e+002	3/10/1.343190e+003
Schwefel	x_0	$(-500,-500,\ldots,-500)$	$(-400,-400,\ldots,-400)$	$(100,100,\ldots,100)$	$(400,400,\ldots,400)$	$(500,500,\ldots,500)$
Dim	10	3/11/4.110923e+003	6/14/4.110923e+003	4/16/4.110923e+003	5/15/4.110923e+003	3/13/4.110923e+003
	100	4/10/4.110923e+004	3/9/4.110929e+004	4/17/4.110923e+004	4/12/4.110923e+004	3/13/4.110923e+004
	300	4/9/1.233278e+005	3/9/1.233279e+005	3/14/1.233284e+005	4/12/1.233277e+005	3/13/1.233277e+005
Schwefel	x_0	$(-500,0,-500,0,\ldots)$	$(-400,0,-400,0,\ldots)$	$(100,0,100,0,\ldots)$	$(400,0,400,0,\ldots)$	$(500,0,500,0,\ldots)$
Dim	10	1/2/4.150492e+003	1/2/4.153457e+003	1/2/4.198244e+003	1/2/4.226201e+003	1/2/4.229166e+003
	100	1/2/4.150492e+004	1/2/4.153457e+004	1/2/4.198244e+004	1/2/4.226201e+004	1/2/4.229166e+004
	300	1/2/1.245147e+005	1/2/1.246037e+005	1/2/1.259473e+005	1/2/1.267860e+005	1/2/1.268750e+005
Griewank	x_0	$(-50,-50,\ldots,-50)$	$(-10,-10,\ldots,-10)$	$(1,1,\ldots,1)$	$(20,20,\ldots,20)$	$(30,30,\ldots,30)$
Dim	10	2/19/7.250909e+000	2/19/1.264953e+000	20/75/1.731886e−006	10/46/6.840017e−007	3/9/5.618497e−006
	100	4/32/2.276173e−002	11/32/6.119425e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	29/82/6.726118e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
Griewank	x_0	$(-50,0,-50,0,\ldots)$	$(-10,0,-10,0,\ldots)$	$(1,0,1,0,\ldots)$	$(20,0,20,0,\ldots)$	$(30,0,30,0,\ldots)$
Dim	10	2/19/4.553192e+000	5/40/1.095684e+000	3/24/2.429039e−001	2/19/1.551065e+000	6/44/1.010824e+000
	100	4/14/1.509048e+000	24/66/4.158154e−006	3/8/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	25/81/2.857231e−005	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000

5. Griewank function.

$$f_{Gri}(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\frac{x_i}{i}, \qquad x_i \in [-600, 600]$$

$$x^* = (0, 0, \ldots, 0), \qquad f_{Gri}(x^*) = 0.$$

The above Benchmark problems can be found at: http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume24/ortizboyer05a-html/node6.html.

The columns of Tables 2–4 have the following meaning:

x_0 : the initial point;

Dim: the dimension of the problem;

NI: the total number of iterations;

NFG: the total number of NF and NG, where $NFG = NF + NG$.

$f(\bar{x})$: denotes the function value at the point $\bar{x}$ when the program is stopped.

Figs. 3 and 4 show that the performance of these methods is relative to *NI* and *NFG* of Tables 2–4, respectively.

Fig. 3 shows that these three methods can solve about 99% of the test problems successfully. Fig. 4 shows the new algorithm and the PRP method outperforms FR method about 1% of the test problems.

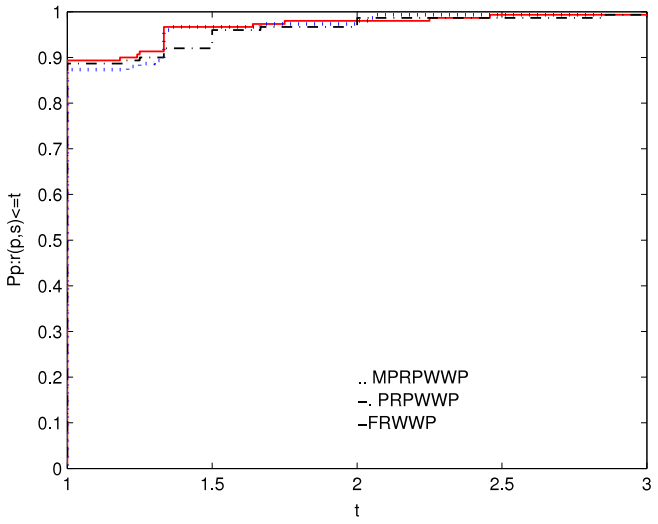

Fig. 3. Performance profiles (NI) of these methods with WWP.

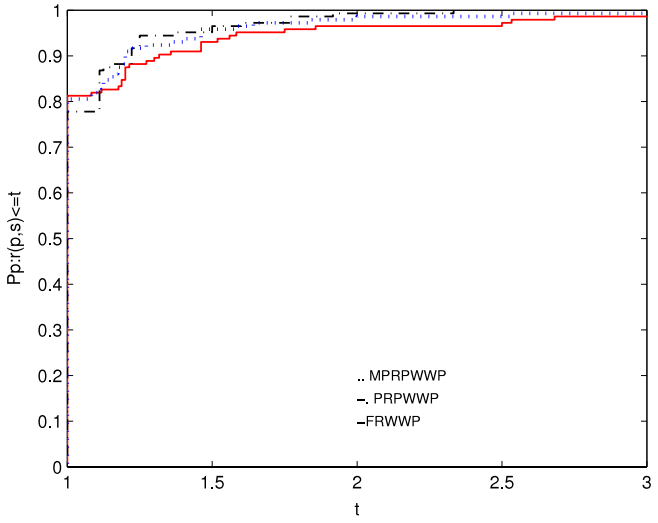

Fig. 4. Performance profiles(NFG) of these methods with WWP.

In summary, the presented numerical results reveal that Algorithm 1, compared with the FR and PRP line search method, has potential advantages for these problems.

5. Conclusions

In this paper, we give a modified conjugate gradient method with the Wolfe–Powell rule for unconstrained optimization problems. The direction d_k has some better properties, especially the sufficient condition (1.8) and condition (3.1) can be hold without any line search technique, and many papers (see [19,35,36] etc) often obtain the two conditions by assumption directly. The global convergence is established, moreover the linear convergence rate with the conjugate gradient method is established for convex functions. The comparison of the numerical results shows that the proposed algorithm is competitive to the other two normal algorithms.

For further research, we should study the convergence of the new algorithm under the other line search rules. Moreover, the choice of the scalars (c_0, μ_1, and μ_2) and more numerical experiments for large practical problems should be done in the future.

Acknowledgements

We are very grateful to the anonymous referees and the editors for their valuable suggestions and comments, which improved our paper greatly.

Table 4
The line search is used by Wolfe–Powell rule. Test results for Algorithm 1.

Sphere	x_0	NI/NFG/$f(\bar{x})$ $(-5,-5,\ldots,-5)$	NI/NFG/$f(\bar{x})$ $(-3,-3,\ldots,-3)$	NI/NFG/$f(\bar{x})$ $(-1,-1,\ldots,-1)$	NI/NFG/$f(\bar{x})$ $(2,2,\ldots,2)$	NI/NFG/$f(\bar{x})$ $(4,4,\ldots,4)$
Dim	10	2/6/7.888609e-030	3/7/0.000000e+000	3/7/1.232595e−031	3/7/4.437343e−030	2/6/1.972152e−030
	100	2/6/6.389773e−027	2/6/1.546167e−026	2/6/7.888609e−029	2/6/1.597443e−027	2/6/1.135960e−026
	300	2/6/6.399240e−025	2/6/6.058452e−026	2/6/3.328007e−027	2/6/5.916457e−027	2/6/1.599810e−025
Sphere	x_0	$(-5,0,-5,0,\ldots)$	$(-3,0,-3,0,\ldots)$	$(-1,0,-1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(4,0,4,0,\ldots)$
Dim	10	2/6/6.310887e−029	3/7/9.860761e−031	3/7/6.162976e−032	3/7/0.000000e+000	3/7/9.860761e−031
	100	2/6/3.944305e−027	2/6/9.860761e−028	3/7/2.218671e−029	2/6/2.465190e−028	2/6/1.577722e−028
	300	2/6/1.431783e−026	2/6/1.848893e−026	2/6/5.990412e−028	2/6/5.798128e−027	2/6/3.833864e−026
Schwefel's	x_0	$(-0.001,\ldots,-0.001)$	$(-0.0002,\ldots,-0.0002)$	$(0.0006,\ldots,0.0006)$	$(0.0007,\ldots,0.0007)$	$(0.0008,\ldots,0.0008)$
Dim	10	3/8/3.539977e−007	3/8/1.415991e−008	3/8/1.274392e−007	3/8/1.734589e−007	3/8/2.265585e−007
	50	5/14/2.297452e−006	4/11/3.471788e−007	5/14/8.270826e−007	5/14/1.125751e−006	5/14/1.470369e−006
	100	7/20/2.554559e−006	5/14/7.557030e−007	6/17/2.317420e−006	6/17/3.154267e−006	7/20/1.634918e−006
Schwefel's	x_0	$(-0.001,0,\ldots)$	$(-0.0002,0,\ldots)$	$(0.0006,0,\ldots)$	$(0.0007,0,\ldots)$	$(0.0008,0,\ldots)$
Dim	10	3/8/6.244149e−007	2/5/5.186065e−008	3/8/2.247894e−007	3/8/3.059633e−007	3/8/3.996256e−007
	50	5/14/3.464888e−006	3/8/6.064742e−007	4/11/1.848353e−006	4/11/2.515814e−006	4/11/3.285961e−006
	100	7/20/6.523845e−006	4/11/9.360044e−007	5/14/3.845945e−006	6/17/3.745873e−006	6/17/4.892569e−006
Rastrigin	x_0	$(-5,-5,\ldots,-5)$	$(-4,-4,\ldots,-4)$	$(1,1,\ldots,1)$	$(2,2,\ldots,2)$	$(3,3,\ldots,3)$
Dim	10	4/13/2.487372e+002	4/13/1.591924e+002	5/16/9.949591e+000	4/13/3.979831e+001	4/13/8.954601e+001
	100	3/9/2.487372e+003	3/9/1.591924e+003	3/9/9.949591e+001	3/9/3.979831e+002	3/9/8.954601e+002
	300	3/9/7.462117e+003	3/9/4.775773e+003	3/8/2.984877e+002	3/8/1.193950e+003	3/9/2.686380e+003
Rastrigin	x_0	$(-5,0,-5,0,\ldots)$	$(-4,0,-4,0,\ldots)$	$(1,0,1,0,\ldots)$	$(2,0,2,0,\ldots)$	$(3,0,3,0,\ldots)$
Dim	10	3/9/1.243686e+002	3/9/7.959622e+001	3/9/4.974795e+000	3/9/1.989916e+001	3/9/4.477301e+001
	100	4/12/1.243686e+003	4/12/7.959622e+002	4/12/4.974795e+001	4/12/1.989916e+002	4/12/4.477301e+002
	300	3/9/3.731058e+003	3/9/2.387887e+003	3/9/1.492439e+002	3/9/5.969747e+002	3/9/1.343190e+003
Schwefel	x_0	$(-500,-500,\ldots,-500)$	$(-400,-400,\ldots,-400)$	$(100,100,\ldots,100)$	$(400,400,\ldots,400)$	$(500,500,\ldots,500)$
Dim	10	3/9/-1.006819e+004	2/6/-4.174987e+003	2/19/3.101787e+003	4/38/4.708237e+003	2/19/5.780458e+002
	100	3/9/-1.006641e+005	2/6/-4.174488e+004	2/19/3.101787e+004	2/19/NaN	2/19/5.780458e+003
	300	3/9/-3.019841e+005	2/6/-1.252317e+005	2/19/9.305360e+004	4/39/5.048414e+005	2/19/1.734137e+004
Schwefel	x_0	$(-500,0,-500,0,\ldots)$	$(-400,0,-400,0,\ldots)$	$(100,0,100,0,\ldots)$	$(400,0,400,0,\ldots)$	$(500,0,500,0,\ldots)$
Dim	10	1/2/5.995721e+003	1/2/5.380480e+002	1/2/3.645808e+003	1/2/7.841610e+003	1/2/2.383937e+003
	100	1/2/5.995721e+004	1/2/5.380480e+003	1/2/3.645808e+004	1/2/7.841610e+004	1/2/2.383937e+004
	300	1/2/1.798716e+005	1/2/1.614144e+004	1/2/1.093742e+005	1/2/2.352483e+005	1/2/7.151812e+004
Griewank	x_0	$(-50,-50,\ldots,-50)$	$(-10,-10,\ldots,-10)$	$(1,1,\ldots,1)$	$(20,20,\ldots,20)$	$(30,30,\ldots,30)$
Dim	10	2/19/7.250909e+000	2/19/1.264953e+000	20/60/8.438297e−007	13/54/1.115419e−006	3/9/5.618497e−006
	100	4/32/2.276173e−002	15/45/7.810149e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	60/164/9.630521e−006	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
Griewank	x_0	$(-50,0,-50,0,\ldots)$	$(-10,0,-10,0,\ldots)$	$(1,0,1,0,\ldots)$	$(20,0,20,0,\ldots)$	$(30,0,30,0,\ldots)$
Dim	10	2/19/4.553192e+000	4/25/9.540580e−001	5/44/1.103014e+000	2/19/1.551065e+000	10/29/1.549244e−006
	100	4/15/1.432188e−014	29/77/7.390224e−006	3/8/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000
	300	51/133/5.541341e−007	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000	2/6/0.000000e+000

References

[1] Y. Dai, A nonmonotone conjugate gradient algorithm for unconstrained optimization, J. Syst. Sci. Complex. 15 (2002) 139–145.
[2] Z. Wei, G. Li, L. Qi, New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems, Appl. Math. Comput. 179 (2006) 407–430.
[3] Z. Wei, S. Yao, L. Liu, The convergence properties of some new conjugate gradient methods, Appl. Math. Comput. 183 (2006) 1341–1350.
[4] G.L. Yuan, X.W. Lu, A new line search method with trust region for unconstrained optimization, Commun. Appl. Nonlinear Anal. 1 (15) (2008) 35–49.
[5] G.L. Yuan, X.W. Lu, Z.X. Wei, New two-point stepsize gradient methods for solving unconstrained optimization problems, Natur. Sci. J. Xiangton Univ. 1 (29) (2007) 13–15.
[6] G.L. Yuan, Z.X. Wei, New line search methods for unconstrained optimization, J. Korean Statist. Soc. 38 (2009) 29–39.
[7] G.L. Yuan, Z.X. Wei, The superlinear convergence analysis of a nonmonotone BFGS algorithm on convex objective functions, Acta Math. Sin. (Engl. Ser.) 1 (24) (2008) 35–42.
[8] G.L. Yuan, Z.X. Wei, Convergence analysis of a modified BFGS method on convex minimizations, Comput. Optim. Appl., doi:10.1007/s10589-008-9219-0.
[9] G.L. Yuan, Z.X. Wei, A rank-one fitting method for unconstrained optimization problems, Math. Appl. 1 (22) (2009) 118–122.
[10] Y. Dai, Y. Yuan, A nonlinear conjugate gradient with a strong global convergence properties, SIAM J. Optim. 10 (2000) 177–182.
[11] R. Fletcher, Practical Method of Optimization, Vol I: Unconstrained Optimization, 2nd ed., Wiley, New York, 1997.
[12] R. Fletcher, C. Reeves, Function minimization bu conjugate gradients, Compute. J. 7 (1964) 149–154.
[13] M.R. Hestenes, E. Stiefel, Method of conjugate gradient for solving linear equations, J, Res. Nat. Bur. Stand. 49 (1952) 409–436.
[14] Y. Liu, C. Storey, Effcient generalized conjugate gradient algorithms, part 1: Theory, J. Optim. Theory Appl. 69 (1992) 17–41.

[15] E. Polak, G. Ribiere, Note sur la xonvergence de directions conjugees, Rev. Francaise informat Recherche Operatinelle, 3e Annee 16 (1969) 35–43.
[16] Y. Dai, Y. Yuan, Nonlinear Conjugate Gradient Methods, Science Press of Shanghai, Shanghai, 2000.
[17] Y. Yuan, W. Sun, Theory and Methods of Optimization, Science Press of China, Beijing, 1999.
[18] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method, in: Lecture Notes in Mathematics, vol. 1066, Spinger-Verlag, Berlin, 1984, pp. 122–141.
[19] J.C. Gibert, J. Nocedal, Global convergence properties of conugate gradient methods for optimization, SIAM J. Optimiz. 2 (1992) 21–42.
[20] Y. Dai, Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, Ann. Oper. Res. 103 (2001) 33–47.
[21] T. Ahmed, D. Storey, Efficient hybrid conjugate gradient techniques, J. Optim. Theory Appl. 64 (1990) 379–394.
[22] A. Al-Baali, Descent property and global convergence of the Flecher–Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985) 121–124.
[23] Y.F. Hu, C. Storey, Global convergence result for conjugate method, J. Optim. Theory Appl. 71 (1991) 399–405.
[24] L. Grippo, S. Lucidi, A globally convergent version of the Polak–Ribière gradient method, Math. Program. 78 (1997) 375–391.
[25] W.W. Hager, H.C. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM J. Optim. 1 (16) (2005) 170–192.
[26] G.H. Yu, Nonlinear self-scaling conjugate gradient methods for large-scale optimization problems, Thesis of Doctor's Degree, Sun Yat-Sen University, 2007.
[27] G.L. Yuan, Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems, Optim. Lett. 3 (2009) 11–21.
[28] G.L. Yuan, X.W. Lu, A modified PRP conjugate gradient method, Ann. Oper. Res. 166 (2009) 73–90.
[29] L. Zhang, W. Zhou, D. Li, A descent modified Polak–Ribière–Polyak conjugate method and its global convergence, IMA J. Numer. Anal. 26 (2006) 629–649.
[30] G. Zoutendijk, Nonlinear programming computational methods, in: J. Abadie (Ed.), Integer and Nonlinear Programming, NorthHolllad, Amsterdam, 1970, pp. 37–86.
[31] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equation in Seveal Variables, Academic Press, 1970.
[32] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1970.
[33] J.J. Moré, B.S. Garbow, K.E. Hillstrome, Testing unconstrained optimization software, ACM Trans. Math. Software. 7 (1981) 17–41.
[34] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.
[35] Z.J. Shi, Convergence of line search methods for unconstrained optimization, Appl. Math. Comput. 157 (2004) 393–405.
[36] H.C. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim 4 (14) (2004) 1043–1056.