RESEARCH ARTICLE

Two New PRP Conjugate Gradient Algorithms for Minimization Optimization Models

Gonglin Yuan^{1,2}, Xiabin Duan^{1*}, Wenjie Liu^{2,3}, Xiaoliang Wang¹, Zengru Cui¹, Zhou Sheng¹

1 Guangxi Colleges and Universities Key Laboratory of Mathematics and Its Applications, College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, P. R. China, **2** School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, P. R. China, **3** Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, P. R. China

* xiabinduan@126.com

OPEN ACCESS

Citation: Yuan G, Duan X, Liu W, Wang X, Cui Z, Sheng Z (2015) Two New PRP Conjugate Gradient Algorithms for Minimization Optimization Models. PLoS ONE 10(10): e0140071. doi:10.1371/journal.pone.0140071

Editor: Yongtang Shi, Nankai University, CHINA

Received: May 27, 2015

Accepted: September 20, 2015

Published: October 26, 2015

Copyright: © 2015 Yuan et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All data are available in the paper.

Funding: This work is supported by China NSF (Grant No. 11261006 and 11161003), NSFC No. 61232016, NSFC No. U1405254, the Guangxi Science Fund for Distinguished Young Scholars (No. 2015GXNSFGA139001) and PAPD issue of Jiangsu advantages discipline.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Two new PRP conjugate Algorithms are proposed in this paper based on two modified PRP conjugate gradient methods: the first algorithm is proposed for solving unconstrained optimization problems, and the second algorithm is proposed for solving nonlinear equations. The first method contains two aspects of information: function value and gradient value. The two methods both possess some good properties, as follows: 1) $\beta_k \geq 0$ 2) the search direction has the trust region property without the use of any line search method 3) the search direction has sufficient descent property without the use of any line search method. Under some suitable conditions, we establish the global convergence of the two algorithms. We conduct numerical experiments to evaluate our algorithms. The numerical results indicate that the first algorithm is effective and competitive for solving unconstrained optimization problems and that the second algorithm is effective for solving large-scale nonlinear equations.

Introduction

As we know, the conjugate gradient method is very popular and effective for solving the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and $g(x)$ denotes the gradient of $f(x)$ at x , the problem [Eq \(1\)](#) also can be applied to model some other problems [\[1–5\]](#). The iterative formula used in the conjugate gradient method is usually given by

$$x_{k+1} = x_k + \alpha_k d_k \quad (2)$$

and

$$d_k = \begin{cases} -g_k & \text{if } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{if } k \geq 2 \end{cases} \quad (3)$$

where $g_k = g(x_k)$, $\beta_k \in \mathbb{R}$ is a scalar, $\alpha_k > 0$ is a step length that is determined by some line search, and d_k denotes the search direction. Different conjugate methods have different choices for β_k . Some of the popular methods [6–12] used to compute β_k are the DY conjugate gradient method [6], FR conjugate gradient method [7], PRP conjugate gradient method [8, 9], HS conjugate gradient method [10], LS conjugate gradient method [11], and CD conjugate gradient method [12]. β_k [8, 9] is defined by

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \quad (4)$$

where $\|\cdot\|$ denotes the Euclidean norm, $y_{k-1} = g_k - g_{k-1}$. The PRP conjugate gradient method is currently considered to have the best numerical performance, but it does not have good convergence. With an exact line search, the global convergence of the PRP conjugate gradient method has been established by Polak and Ribière [8] for convex objective functions. However, Powell [13] proposed a counter example that proved the existence of nonconvex functions on which the PRP conjugate gradient method does not have global convergence, even with an exact line search. With the weak Wolfe-Powell line search, Gilbert and Nocedal [14] proposed a modified PRP conjugate gradient method by restricting β_k to be not less than zero and proved that it has global convergence, with the hypothesis that it satisfies the sufficient descent condition. Gilbert and Nocedal [14] also gave an example showing that β_k may be negative even though the objective function is uniformly convex. When the Strong Wolfe-Powell line search was used, Dai [15] gave an example showing that the PRP method cannot guarantee that every step search direction is the descent direction, even if the objective function is uniformly convex.

Through the above observations and [13, 14, 16–18], we know that the following sufficient descent condition

$$-g_k^T d_k \geq b \|g_k\|^2, \forall b > 0 \quad (5)$$

and the condition β_k is not less than zero are very important for establishing the global convergence of the conjugate gradient method.

The weak Wolfe-Powell (WWP) line search is designed to compute α_k and is usually used for the global convergence analysis. The WWP line search is as follows

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k g_k^T d_k \quad (6)$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \delta_2 g_k^T d_k \quad (7)$$

where $\delta_1 \in (0, \frac{1}{2})$, $\delta_2 \in (\delta_1, 1)$.

Recently, many new conjugate gradient methods ([19–28] etc.) that possess some good properties have been proposed for solving unconstrained optimization problems.

In Section 2, we state the motivation behind our approach and give a new modified PRP conjugate gradient method and new algorithm for solving problem Eq (1). In Section 3, we prove that the search direction of our new algorithm satisfies the sufficient descent property and trust region property; moreover, we establish the global convergence of the new

algorithm with the WWP line search. In Section 4, we provide numerical experiment results for some test problems.

New algorithm for unconstrained optimization

Wei et al. [29] give a new PRP conjugate gradient method usually called the WYL method. When the WWP line search is used, this WYL method has global convergence under the sufficient descent condition. Zhang [30] give a modified WYL method called the NPRP method as follows

$$\beta_k^{NPRP} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}|}{\|g_{k-1}\|^2}$$

The NPRP method possesses better convergence properties. The above formula for y_{k-1} contains only gradient value information, but some new y_{k-1} formulas [31, 32] contain information on gradient value and function value. Yuan et al. [32] propose a new y_{k-1} formula as follows

$$y_{k-1}^m = y_{k-1} + \frac{\max\{\rho_{k-1}, 0\}}{\|s_{k-1}\|^2} s_{k-1},$$

and

$$\rho_{k-1} = 2 [f(x_{k-1}) - f(x_k)] + (g(x_k) + g(x_{k-1}))^T s_{k-1}.$$

Where $s_{k-1} = x_k - x_{k-1}$.

Li and Qu [33] give a modified PRP conjugate method as follows

$$\beta_k = \frac{g_k^T y_{k-1}}{\max\{t \|d_{k-1}\|, \|g_{k-1}\|^2\}}, \quad t > 0$$

and

$$d_k = -g_k - \beta_k \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k + \beta_k d_{k-1}, \quad d_0 = -g_0.$$

Under suitable conditions, Li and Qu [33] prove that the modified PRP conjugate method has global convergence.

Motivated by the above discussions, we propose a new modified PRP conjugate method as follows

$$\beta_k^{BPRP} = \frac{\min\left\{|g_k^T y_{k-1}^m|, u_1 \left(\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}|\right)\right\}}{u_2 \|d_{k-1}\| \|y_{k-1}\| + \|g_{k-1}\|^2} \quad (8)$$

and

$$d_k = \begin{cases} -g_k & \text{if } k = 1 \\ -g_k - \beta_k^{BPRP} \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k + \beta_k^{BPRP} d_{k-1} & \text{if } k \geq 2 \end{cases} \quad (9)$$

where $u_1 > 0$, $u_2 > 0$, y_{k-1}^m is the y_{k-1}^m of [32].

As $\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}| \geq 0$, it follows directly from the above formula that $\beta_k^{BPRP} \geq 0$.

Next, we present a new algorithm and its diagram (Fig 1) as follows.

Fig 1. The diagram about Algorithm 2.1.

doi:10.1371/journal.pone.0140071.g001

Algorithm 2.1

Step 0: Given the initial point $x_1 \in \mathbb{R}^n$, $u_1 > 0$, $u_2 > 0$, $\varepsilon_1 \geq 0$, $0 < \delta_1 < \frac{1}{2}$, $\delta_1 < \delta_2 < 1$, set $d_1 = -\nabla f(x_1) = -g_1$, $k := 1$.

Step 1: Calculate $\|g_k\|$; if $\|g_k\| \leq \varepsilon_1$, stop; otherwise, go to step 2.

Step 2: Calculate step length α_k by the WWP line search.

Step 3: Set $x_{k+1} = x_k + \alpha_k d_k$, then calculate $\|g_{k+1}\|$; if $\|g_{k+1}\| \leq \varepsilon_1$, stop; otherwise, go to step 4.

Step 4: Calculate the scalar β_{k+1} by Eq (8) and calculate the search direction d_{k+1} by Eq (9).

Step 5: Set $k := k + 1$; go to step 2.

Global convergence analysis

Some suitable assumptions are often used to analyze the global convergence of the conjugate gradient method. Here, we state it as follows

Assumption 3.1

1. The level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_1)\}$ is bounded.
2. In some neighborhood H of Ω , f is a continuously differentiable function, and the gradient function g of f is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \forall x, y \in H \quad (10)$$

By Assumption 3.1, it is easy to obtain that there exist two constants $A > 0$ and $\eta_1 > 0$ satisfying

$$\|x\| \leq A, \|g(x)\| \leq \eta_1, \forall x \in \Omega \quad (11)$$

Lemma 0.1 Let the sequence $\{d_k\}$ be generated by Eq (9); then, we have

$$g_k^T d_k = -\|g_k\|^2, \forall k \geq 1 \quad (12)$$

Proof When $k = 1$, we can obtain $g_1^T d_1 = -\|g_1\|^2$ by Eq (9), so Eq (12) holds. When $k \geq 2$, we can obtain

$$\begin{aligned} g_k^T d_k &= g_k^T (-g_k - \beta_k^{BPRP} \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k + \beta_k^{BPRP} d_{k-1}) \\ &= -\|g_k\|^2 \end{aligned}$$

The proof is achieved.

We know directly from above Lemma that our new method has the sufficient descent property.

Lemma 0.2 Let the sequence $\{x_k\}$ and $\{d_k, g_k\}$ be generated by Algorithm 2.1, and suppose that Assumption 3.1 holds; then, we can obtain

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \quad (13)$$

Proof By Eq (7) and the Cauchy-Schwarz inequality, we have

$$-(1 - \delta_2)g_k^T d_k \leq \|g_{k+1} - g_k\| \|d_k\|$$

Combining the above inequality with Assumption 3.1 ii) generates

$$-(1 - \delta_2)g_k^T d_k \leq L\alpha_k \|d_k\|^2$$

it is easy to know $g_k^T d_k \leq 0$ by lemma 0.1. By combining the above inequality with Eq (6), we obtain

$$f_k - f_{k+1} \geq \frac{\delta_1(1 - \delta_2)}{L} \frac{(g_k^T d_k)^2}{\|d_k\|^2}.$$

Summing up the above inequalities from $k = 1$ to $k = \infty$, we can deduce that

$$\frac{\delta_1(1 - \delta_2)}{L} \sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \leq f_1 - f_{\infty}.$$

By Eq (6), Assumption 3.1 and lemma 0.1, we know that $\{f_k\}$ is bounded below, so we obtain

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty.$$

This finishes the proof.

The Eq (13) is usually called the Zoutendijk condition [34], and it is very important for establishing global convergence.

Lemma 0.3 Let the sequence $\{\beta_k, d_k\}$ be generated by Algorithm 2.1, we have

$$\|d_k\| \leq N \|g_k\| \quad (14)$$

where $N = 1 + \frac{4u_1}{u_2}$.

Proof When $d_k = 0$, we directly get $g_k = 0$ from Eq (12). When $d_k \neq 0$, by the Cauchy-Schwarz inequality, we can easily obtain

$$\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}| \leq g_k^T (g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})$$

and

$$\begin{aligned} g_k^T (g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1}) &\leq \|g_k\| \|(g_k - g_{k-1}) + (g_{k-1} - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})\| \\ &\leq 2 \|g_k\| \|g_k - g_{k-1}\| \end{aligned}$$

We can obtain

$$\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}| \leq 2 \|g_k\| \|y_{k-1}\|$$

Using Eq (8), we have

$$\begin{aligned} |\beta_k^{BPRP}| &\leq \frac{u_1 \left(\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}| \right)}{u_2 \|d_{k-1}\| \|y_{k-1}\|} \\ &\leq \frac{2u_1 \|g_k\|}{u_2 \|d_{k-1}\|} \end{aligned}$$

Finally, when $k \geq 2$ by Eq (9), we have

$$\begin{aligned} \|d_k\| &\leq \|g_k\| + |\beta_k^{BPRP}| \frac{\|g_k\| \|d_{k-1}\|}{\|g_k\|^2} \|g_k\| + |\beta_k^{BPRP}| \|d_{k-1}\| \\ &\leq \|g_k\| + \frac{2u_1}{u_2} \|g_k\| + \frac{2u_1}{u_2} \|g_k\| \\ &\leq (1 + \frac{4u_1}{u_2}) \|g_k\| \end{aligned}$$

Let $N = 1 + \frac{4u_1}{u_2}$; we obtain $\|d_k\| \leq N \|g_k\|$. This finishes the proof.

This lemma also shows that the search direction of our algorithm has the trust region property.

Theorem 0.1 Let the sequence $\{d_k, g_k, \beta_k\}$ and $\{x_k\}$ be generated by Algorithm 2.1. Suppose that Assumption 3.1 holds; then

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad (15)$$

Proof By Eqs (12) and (13), we obtain

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty \quad (16)$$

By Eq (14), we have $\|d_k\|^2 \leq N^2 \|g_k\|^2$; then, we obtain

$$\|g_k\|^2 \leq N^2 \frac{\|g_k\|^4}{\|d_k\|^2},$$

which together with Eq (16) can yield

$$\sum_{k=1}^{\infty} \|g_k\|^2 \leq N^2 \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty.$$

From the above inequality, we can obtain $\lim_{k \rightarrow \infty} \|g_k\| = 0$. The proof is finished.

Numerical Results

When β_{k+1} and d_{k+1} are calculated by Eqs (4) and (3), respectively, in step 4 of Algorithm 2.1, we call it the PRP conjugate gradient algorithm. We test Algorithm 2.1 and the PRP conjugate gradient algorithm using some benchmark problems. The test environment is MATLAB 7.0, on a Windows 7 system. The initial parameters are given by

$$u_1 = 1, u_2 = 2, \delta_1 = 0.2, \delta_2 = 0.8, \varepsilon_1 = 10^{-6}.$$

We use the following Himmeblau stop rule, which satisfies

If $|f(x_k)| \leq \varepsilon_2$, let $stop1 = stop1 = |f(x_k) - f(x_{k+1})|$; otherwise, let $stop1 = \frac{|f(x_k) - f(x_{k+1})|}{|f(x_k)|}$. The test program will be stopped if $stop1 < \varepsilon_3$ or $\|g(x_k)\| < \varepsilon_1$ is satisfied, where $\varepsilon_2 = \varepsilon_3 = 10^{-6}$. When the total number of iterations is greater than one thousand, the test program will be stopped. The test results are given in Tables 1 and 2: x_1 denotes the initial point, Dim denotes the dimension of test function, NI denotes the the total number of iterations, and NFG = NF+NG (NF and NG denote the number of the function evaluations and the number of the gradient evaluations, respectively). f' denotes the function value when the program is stopped. The test problems are defined as follows.

Table 1. Test results for Algorithm 2.1.

Problems	Dim	x_1	NI/NFG	f'
1	50	(-426,-426,...,-426)	2/9	6.363783e-004
	120	(-426,-426,...,-426)	2/9	1.527308e-003
	200	(-426,-426,...,-426)	2/9	2.545514e-003
	1000	(-410,-410,...,-410)	3/12	1.272757e-002
2	50	(3,3,...,3)	0/2	-1.520789e-060
	120	(5,5,...,5)	0/2	0.000000e+000
	200	(6,6,...,6)	0/2	0.000000e+000
	1000	(1,1,...,1)	0/2	-7.907025e-136
3	50	(-0.00001,0,-0.00001,0,...)	2/8	1.561447e-009
	120	(-0.00001,0,-0.00001,0,...)	2/8	1.769900e-008
	200	(-0.00001,0,-0.00001,0,...)	2/8	7.906818e-008
	1000	(0.000001,0,0.000001,0,...)	2/8	9.619586e-008
4	50	(-4,-4,...,-4)	1/6	1.577722e-028
	120	(-2,-2,...,-2)	1/6	3.786532e-028
	200	(1,1,...,1)	1/6	7.730837e-027
	1000	(3,3,...,3)	1/6	1.079951e-024
5	50	(-7,0,-7,0,...)	2/10	0.000000e+000
	120	(0.592,0,0.592,0,...)	4/14	3.183458e-007
	200	(0.451,0,0.451,0,...)	4/14	3.476453e-007
	1000	(0.38,0,0.38,0,...)	1/6	0.000000e+000
6	50	(1.001,1.001,...,1.001)	2/36	4.925508e-003
	120	(1.001,1.001,...,1.001)	2/36	1.198551e-002
	200	(1.001,1.001,...,1.001)	2/36	2.006158e-002
	1000	(1.001,1.001,...,1.001)	2/36	1.009107e-001
7	50	(0.01,0,0.01,0,...)	0/2	3.094491e-002
	120	(-0.05,0,-0.05,0,...)	0/2	2.066363e-001
	200	(0.01,0,0.01,0,...)	0/2	3.094491e-002
	1000	(0.07,0,0.07,0,...)	0/2	3.233371e-001
8	50	(0.003,0.003,...,0.003)	3/26	0.000000e+000
	120	(0.005,0.005,...,0.005)	2/9	0.000000e+000
	200	(0.006,0,0.006,0,...)	2/9	0.000000e+000
	1000	(0.015,0.015,...,0.015)	2/8	0.000000e+000

doi:10.1371/journal.pone.0140071.t001

1. Schwefel function:

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

$$x^* = (-420.9687, -420.9687, \dots, -420.9687), f_{Sch}(x^*) = 0.$$

Table 2. Test results for the PRP conjugate gradient algorithm.

Problems	Dim	x_1	NI/NFG	f'
1	50	(-426,-426,...,-426)	2/24	6.363783e-004
	120	(-426,-426,...,-426)	2/11	1.527308e-003
	200	(-426,-426,...,-426)	3/41	2.545514e-003
	1000	(-410,-410,...,-410)	3/41	1.272757e-002
2	50	(3,3,...,3)	0/2	-1.520789e-060
	120	(5,5,...,5)	0/2	0.000000e+000
	200	(6,6,...,6)	0/2	0.000000e+000
	1000	(1,1,...,1)	0/2	-7.907025e-136
3	50	(-0.00001,0,-0.00001,0,...)	2/8	1.516186e-009
	120	(-0.00001,0,-0.00001,0,...)	2/8	1.701075e-008
	200	(-0.00001,0,-0.00001,0,...)	2/8	7.579825e-008
	1000	(0.000001,0,0.000001,0,...)	2/8	9.198262e-008
4	50	(-4,-4,...,-4)	1/6	1.577722e-028
	120	(-2,-2,...,-2)	1/6	3.786532e-028
	200	(1,1,...,1)	1/6	7.730837e-027
	1000	(3,3,...,3)	1/6	1.079951e-024
5	50	(-7,0,-7,0,...)	4/16	3.597123e-013
	120	(0.592,0,0.592,0,...)	5/17	3.401145e-007
	200	(0.451,0,0.451,0,...)	5/17	4.566281e-007
	1000	(0.38,0,0.38,0,...)	1/6	0.000000e+000
6	50	(1.001,1.001,...,1.001)	2/36	4.925508e-003
	120	(1.001,1.001,...,1.001)	2/36	1.198551e-002
	200	(1.001,1.001,...,1.001)	2/36	2.006158e-002
	1000	(1.001,1.001,...,1.001)	2/36	1.009107e-001
7	50	(0.01,0,0.01,0,...)	0/2	3.094491e-002
	120	(-0.05,0,-0.05,0,...)	0/2	2.066363e-001
	200	(0.01,0,0.01,0,...)	0/2	3.094491e-002
	1000	(0.07,0,0.07,0,...)	0/2	3.233371e-001
8	50	(0.003,0.003,...,0.003)	2/10	0.000000e+000
	120	(0.005,0.005,...,0.005)	2/10	0.000000e+000
	200	(0.006,0,0.006,0,...)	2/10	0.000000e+000
	1000	(0.015,0.015,...,0.015)	2/22	3.636160e-009

doi:10.1371/journal.pone.0140071.t002

2. Langerman function:

$$f_{Lan}(x) = - \sum_{i=1}^m c_i e^{-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2} \cos \left(\pi \sum_{j=1}^n (x_j - a_{ij})^2 \right), \quad x_i \in [0, 10], \quad m = n,$$

$$x^* = random, \quad f_{Lan}(x^*) = random.$$

3. Schwefel's function

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

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

4. Sphere function:

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

$$x^* = (0, 0, \dots, 0), f_{Sph}(x^*) = 0.$$

5. Griewangk function:

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

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

6. Rosenbrock function:

$$f_{Ros}(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], x_i \in [-2.048, 2.048],$$

$$x^* = (1, \dots, 1), f_{Ros}(x^*) = 0.$$

7. Ackley function:

$$f_{Ack}(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}, x_i \in [-30, 30],$$

$$x^* = (0, 0, \dots, 0), f_{Ack}(x^*) = 0.$$

8. Rastrigin function:

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

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

Fig 2. Performance profiles of the two algorithms (NI).

doi:10.1371/journal.pone.0140071.g002

It is easy to see that the two algorithms are effective for the above eight test problems listed in Tables 1 and 2. We use the tool of Dolan and Moré [35] to analyze the numerical performance of the two algorithms.

For the above eight test problems, Fig 2 shows the numerical performance of the two algorithms when the information of NI is considered, and Fig 3 shows the numerical performance of the two algorithms when the information of NFG is considered. From the above two figures, it is easy to see that Algorithm 2.1 yields a better numerical performance than the PRP conjugate gradient algorithm on the whole. From Tables 1 and 2 and the two figures, we can conclude that Algorithm 2.1 is effective and competitive for solving unconstrained optimization problems.

A new algorithm is given for solving nonlinear equations in the next section. The sufficient descent property and the trust region property of the new algorithm are proved in Section 6; moreover, we establish the global convergence of the new algorithm. In Section 7, the numerical results are presented.

New algorithm for nonlinear equations

We consider the system of nonlinear equations

$$q(x) = 0, \quad x \in \mathbb{R}^n. \quad (17)$$

where $q: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuously differentiable and monotonic function. $\nabla q(x)$ denotes the Jacobian matrix of $q(x)$; if $\nabla q(x)$ is symmetric, we call Eq (17) symmetric nonlinear equations. As $q(x)$ is monotonic, the following inequality

$$(q(x) - q(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n$$

Fig 3. Performance profiles of the two algorithms (NFG).

doi:10.1371/journal.pone.0140071.g003

holds. If a norm function is defined as follows

$$h(x) = \frac{1}{2} \|q(x)\|^2$$

and we define the unconstrained optimization problem as follows,

$$\min h(x), \quad x \in \mathbb{R}^n \quad (18)$$

We know directly that the problem [Eq \(17\)](#) is equivalent to the problem [Eq \(18\)](#).

The iterative formula [Eq \(2\)](#) is also usually used in many algorithms for solving problem [Eq \(17\)](#). Many algorithms ([36–41], etc.) have been proposed for solving special classes of nonlinear equations. We are more interested in the process of dealing with large-scale nonlinear equations. By [Eq \(2\)](#), it is easy to see that the two factors of stepsize α_k and search direction d_k are very important for dealing with large-scale problems. When dealing with large-scale nonlinear equations and unconstrained optimization problems, there are many popular methods ([38, 42–46] etc.) for computing d_k , such as conjugate gradient methods, spectral gradient methods, and limited-memory quasi-Newton approaches. Some new line search methods [37, 47] have been proposed for calculating α_k . Li and Li [48] provide the following new derivative-free line search method

$$-q(x_k + \alpha_k d_k)^T d_k \geq \sigma_3 \alpha_k \|q(x_k + \alpha_k d_k)\| \|d_k\|^2, \quad (19)$$

where $\alpha_k = \max\{\gamma, \rho\gamma, \rho^2\gamma, \dots\}$, $\rho \in (0, 1)$, $\sigma_3 > 0$ and $\gamma > 0$. This line search method is very effective for solving large-scale nonlinear monotonic equations.

Solodov and Svaiter [49] presented a hybrid projection-proximal point algorithm that could conquer some drawbacks when the form [Eq \(18\)](#) is used with nonlinear equations. Yuan et al.

[50] proposed a three-term PRP conjugate gradient algorithm by using the projection-based technique, which was introduced by Solodov et al. [51] for optimization problems. The projection-based technique is very effective for solving nonlinear equations. It involves certain methods to compute search direction d_k and certain line search methods to calculate α_k , which satisfies

$$q(w_k)^T(x_k - w_k) > 0$$

in which $w_k = x_k + \alpha_k d_k$. For any x^* that satisfies $q(x^*) = 0$, considering that $q(x)$ is monotonic, we can obtain

$$q(w_k)^T(x^* - w_k) \leq 0.$$

Thus, it is easy to obtain the current iterate x_k , which is strictly separated from the zeros of the system of equations Eq (17) by the following hyperplane

$$T_k = \{x \in \mathbb{R}^n \mid q(w_k)^T(x - w_k) = 0\}$$

Then, the iterate x_{k+1} can be obtained by projecting x_k onto the above hyperplane. The projection formula can be set as follows

$$x_{k+1} = x_k - \frac{q(w_k)^T(x_k - w_k)}{\|q(w_k)\|^2} q(w_k) \quad (20)$$

Yuan et al. [50] present a three-term Polak-Ribière-Polyak conjugate gradient algorithm in which the search direction d_k is defined as follows

$$d_k = \begin{cases} -q_k & \text{if } k = 0 \\ -q_k + \frac{q_k^T y_{k-1} d_{k-1} - q_k^T d_{k-1} y_{k-1}}{\max\{\mu \|d_{k-1}\| \|y_{k-1}\|, \|q_{k-1}\|^2\}} & \text{if } k \geq 1 \end{cases}$$

where $y_{k-1} = q_k - q_{k-1}$. The derivative-free line search method [48] and the projection-based techniques are used by the algorithm [50], proved to be very suitable for solving large-scale nonlinear equations. The most attractive property of algorithm [50] is the trust region property of d_k .

Motivated by our new modified PRP conjugate gradient formula, proposed in Section 2, we proposed the following modified PRP conjugate gradient formula

$$\beta_k^* = \frac{\min\left\{|q_k^T(q_k - q_{k-1})|, u_3\left(\|q_k\|^2 - \frac{\|q_k\|}{\|q_{k-1}\|} |q_k^T q_{k-1}|\right)\right\}}{u_4 \|d_{k-1}\| \|q_k - q_{k-1}\| + \|q_{k-1}\|^2} \quad (21)$$

and

$$d_k = \begin{cases} -q_k, & \text{if } k = 1 \\ -q_k - \beta_k^* \frac{q_k^T d_{k-1}}{\|q_k\|^2} q_k + \beta_k^* d_{k-1}, & \text{if } k \geq 2 \end{cases} \quad (22)$$

Where $u_3 > 0$, $u_4 > 0$. It is easy to see that $\beta_k^* \geq 0$, motivated by the above observation and [50]. We present a new algorithm for solving problem Eq (17): it uses our modified PRP conjugate gradient formula Eqs (21) and (22). Here, we list the new algorithm and its diagram (Fig 4) as follows.

Fig 4. The diagram about Algorithm 5.1.

doi:10.1371/journal.pone.0140071.g004

Algorithm 5.1

Step 1: Given the initial point $x_1 \in \mathbb{R}^n$, $\varepsilon_4 > 0$, $\rho \in (0, 1)$, $\sigma_3 > 0$, $\gamma > 0$, $u_3 > 0$, $u_4 > 0$, and $k := 1$.
Step 2: If $\|q(x_k)\| \leq \varepsilon_4$, stop; otherwise, go to step 3.
Step 3: Compute d_k by Eq (22) and calculate α_k by Eq (19).
Step 4: Set the next iterate to be $w_k = x_k + \alpha_k d_k$.
Step 5: If $\|q(w_k)\| \leq \varepsilon_4$, stop and set $x_{k+1} = w_k$; otherwise, calculate x_{k+1} by Eq (20).
Step 6: Set $k := k + 1$; go to step 2.

Convergence Analysis

When we analyze the global convergence of Algorithm 5.1, we require the following suitable assumptions.

Assumption 6.1

1. The solution set of the problem Eq (17) is nonempty.
2. $q(x)$ is Lipschitz continuous, namely, there exists a constant $E > 0$ such that

$$\|q(x) - q(y)\| \leq E \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

By Assumption 6.1, it is easy to obtain that there exists a positive constant ζ that satisfies

$$\|q(x_k)\| \leq \zeta \quad (23)$$

Lemma 0.4 Let the sequence $\{d_k\}$ be generated by Eq (22); then, we can obtain

$$q_k^T d_k = -\|q_k\|^2 \quad (24)$$

and

$$\|q_k\| \leq \|d_k\| \leq \left(1 + \frac{4u_3}{u_4}\right) \|q_k\| \quad (25)$$

Proof As the proof is similar to Lemma 0.1 and Lemma 0.3 of this paper, we omit it here. Similar to Lemma 3.1 of [50] and theorem 2.1 of [51], it is easy to obtain the following lemma. Here, we omit this proof and only list it.

Lemma 0.5 Suppose that Assumption 6.1 holds and x^* is a solution of problem Eq (17) that satisfies $g(x^*) = 0$. Let the sequence $\{x_k\}$ be obtained by Algorithm 5.1; then, the $\{x_k\}$ is a bounded sequence and

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2$$

holds. Moreover, either $\{x_k\}$ is a infinite sequence and

$$\sum_{k=1}^{\infty} \|x_{k+1} - x_k\|^2 < \infty$$

or the $\{x_k\}$ is a finite sequence and a solution of problem Eq (17) is the last iteration.

Lemma 0.6 Suppose that Assumption 6.1 holds, then, an iteration $x_{k+1} = x_k + \alpha_k d_k$ will be generated by Algorithm 5.1 in a finite number of backtracking steps.

Proof We will obtain this conclusion by contradiction: suppose that $\|q_k\| \rightarrow 0$ does not hold; then, there exists a positive constant ε_5 that satisfies

$$\|q_k\| \geq \varepsilon_5, \quad \forall k \geq 1 \quad (26)$$

suppose that there exist some iterate indexes k' that do not satisfy the condition Eq (19). We let $\alpha_{k'}^{(c)} = \rho^{(c)}\gamma$ then it can obtain

$$-q(x_{k'} + \alpha_{k'}^{(c)} d_{k'})^T d_{k'} < \sigma_3 \alpha_{k'}^{(c)} \|q(x_{k'} + \alpha_{k'}^{(c)} d_{k'})\| \|d_{k'}\|^2, \quad \forall c \in N^* \cup \{0\}.$$

By Assumption 6.1 (b) and Eq (24), we find

$$\begin{aligned} \|d_{k'}\|^2 &= -q_{k'}^T d_{k'} \\ &= [q(x_{k'} + \alpha_{k'}^{(c)} d_{k'}) - q(x_{k'})]^T d_{k'} - q(x_{k'} + \alpha_{k'}^{(c)} d_{k'})^T d_{k'} \\ &< [E + \sigma_3 \|q(x_{k'} + \alpha_{k'}^{(c)} d_{k'})\|] \alpha_{k'}^{(c)} \|d_{k'}\|^2 \end{aligned}$$

By Eqs (23) and (25), we can obtain

$$\begin{aligned} \|q(x_{k'} + \alpha_{k'}^{(c)} d_{k'})\| &\leq \|q(x_{k'} + \alpha_{k'}^{(c)} d_{k'}) - q_{k'}\| + \|q_{k'}\| \\ &\leq E \alpha_{k'}^{(c)} \|d_{k'}\| + \zeta \\ &\leq E \gamma \zeta \left(1 + \frac{4u_3}{u_4}\right) + \zeta \end{aligned}$$

Thus, we obtain

$$\alpha_{k'}^{(c)} > \frac{\varepsilon_5^2 u_4^2}{\left[E + \sigma_3 \left(E \gamma \zeta \left(1 + \frac{4u_3}{u_4}\right) + \zeta\right)\right] (u_4 + 4u_3)^2 \zeta^2}, \quad \forall c \in N^* \cup \{0\}$$

which shows that $\alpha_{k'}^{(c)}$ is bounded below. This contradicts the definition of $\alpha_{k'}^{(c)}$; so, the lemma holds.

Similar to Theorem 3.1 of [50], we list the following theorem but omit its proof.

Theorem 0.2 Let the sequence $\{x_{k+1}, q_{k+1}\}$ and $\{\alpha_k, d_k\}$ be generated by Algorithm 5.1. Suppose that Assumption 6.1 holds; then, we have

$$\liminf_{k \rightarrow \infty} \|q_k\| = 0. \quad (27)$$

Numerical results

When the following d_k formula of the famous PRP conjugate gradient method [8, 9]

$$d_k = \begin{cases} -q_k & \text{if } k = 1 \\ -q_k + \frac{q_k^T (q_k - q_{k-1})}{\|q_{k-1}\|^2} d_{k-1} & \text{if } k \geq 2 \end{cases}$$

is used to compute d_k in step 3 of Algorithm 5.1, then it is called PRP algorithm. We test Algorithm 5.1 and the PRP algorithm for some problems in this section. The test environment is

Table 3. Test results for Algorithm 5.1.

Function	Dim	NI/NG	cputime	GF
1	3000	55/209	2.043613	9.850811e-006
	5000	8/33	0.858005	6.116936e-006
	30000	26/127	100.792246	8.983556e-006
	45000	7/36	62.681202	7.863794e-006
	50000	5/26	56.659563	5.807294e-006
2	3000	43/86	1.076407	8.532827e-006
	5000	42/84	2.745618	8.256326e-006
	30000	38/76	73.039668	8.065468e-006
	45000	37/74	164.284653	8.064230e-006
	50000	36/72	201.288090	9.519786e-006
3	3000	5/6	0.093601	1.009984e-008
	5000	5/6	0.249602	6.263918e-009
	30000	18/33	32.775810	2.472117e-009
	45000	21/39	91.229385	2.840234e-010
	50000	21/39	108.202294	2.661223e-010
4	3000	95/190	2.137214	9.497689e-006
	5000	97/194	5.834437	9.048858e-006
	30000	103/206	194.954450	8.891642e-006
	45000	104/208	446.568463	9.350859e-006
	50000	104/208	549.529123	9.856874e-006
5	3000	64/128	1.497610	9.111464e-006
	5000	65/130	4.102826	9.525878e-006
	30000	70/140	132.117247	8.131796e-006
	45000	70/140	297.868309	9.959279e-006
	50000	71/142	374.964004	8.502923e-006
6	3000	1/2	0.031200	0.000000e+000
	5000	1/2	0.062400	0.000000e+000
	30000	1/2	1.918812	0.000000e+000
	45000	1/2	4.258827	0.000000e+000
	50000	1/2	5.194833	0.000000e+000
7	3000	35/71	0.842405	9.291878e-006
	5000	34/69	2.121614	8.658237e-006
	30000	30/61	58.391174	8.288490e-006
	45000	29/59	135.627269	8.443996e-006
	50000	29/58	153.801386	9.993530e-006
8	3000	0/1	0.015600	0.000000e+000
	5000	0/1	0.046800	0.000000e+000
	30000	0/1	1.326008	0.000000e+000
	45000	0/1	2.917219	0.000000e+000
	50000	0/1	3.510022	0.000000e+000

doi:10.1371/journal.pone.0140071.t003

MATLAB 7.0 on a Windows 7 system. The initial parameters are given by

$$\sigma_3 = u_4 = 0.02, \gamma = 1, \rho = 0.1, u_3 = 1, \varepsilon_4 = 10^{-5}.$$

When the number of iterations is greater than or equal to one thousand and five hundred, the test program will also be stopped. The test results are given in Tables 3 and 4. As we know,

Table 4. Test results for PRP algorithm.

Function	Dim	NI/NG	cputime	GF
1	3000	58/220	2.043613	9.947840e-006
	5000	24/97	2.496016	9.754454e-006
	30000	29/141	109.668703	9.705424e-006
	45000	13/66	118.108357	9.450575e-006
	50000	10/51	112.383120	9.221806e-006
2	3000	48/95	1.138807	8.647042e-006
	5000	46/91	2.932819	9.736889e-006
	30000	41/81	78.733705	9.983531e-006
	45000	40/79	181.709965	9.632281e-006
	50000	40/79	212.832164	9.121412e-006
3	3000	11/12	0.171601	1.012266e-008
	5000	11/12	0.530403	8.539532e-009
	30000	23/38	39.749055	2.574915e-009
	45000	26/44	100.542645	2.931611e-010
	50000	26/44	123.864794	2.838473e-010
4	3000	104/208	2.246414	9.243312e-006
	5000	106/212	6.193240	9.130520e-006
	30000	113/226	219.821009	8.747379e-006
	45000	114/228	487.908728	9.368026e-006
	50000	114/228	611.976323	9.874918e-006
5	3000	35/53	0.561604	2.164559e-006
	5000	35/53	1.716011	1.291210e-006
	30000	35/53	55.926358	1.336971e-006
	45000	33/49	116.361146	2.109293e-006
	50000	33/49	147.452145	2.225071e-006
6	3000	1/2	0.031200	0.000000e+000
	5000	1/2	0.062400	0.000000e+000
	30000	1/2	1.965613	0.000000e+000
	45000	1/2	4.290028	0.000000e+000
	50000	1/2	5.257234	0.000000e+000
7	3000	40/80	0.904806	9.908999e-006
	5000	39/78	2.386815	9.198351e-006
	30000	34/68	66.440826	9.515010e-006
	45000	33/66	140.026498	9.366998e-006
	50000	33/66	173.597913	8.886013e-006
8	3000	0/1	0.015600	0.000000e+000
	5000	0/1	0.031200	0.000000e+000
	30000	0/1	1.279208	0.000000e+000
	45000	0/1	2.808018	0.000000e+000
	50000	0/1	3.432022	0.000000e+000

doi:10.1371/journal.pone.0140071.t004

when the line search cannot guarantee that d_k satisfies $q_k^T d_k < 0$, some uphill search direction may be produced; the line search method possibly fails in this case. In order to prevent this situation, when the search time is greater than or equal to fifteen in the inner cycle of our program, we set α_k that is acceptable. NG, NI stand for the number of gradient evaluations and iterations respectively. Dim denotes the dimension of the testing function, and cputime denotes

the cpu time in seconds. GF denotes the evaluation of the final function norm when the program terminates. The test functions all have the following form

$$q(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$$

the concrete function definitions are given as follows.

Function 1. Exponential function 2

$$\begin{aligned} f_1(x) &= e^{x_1} - 1, \\ f_i(x) &= \frac{i}{10}(e^{x_i} + x_{i-1} - 1), \quad i = 2, 3, \dots, n \end{aligned}$$

$$\text{Initial guess: } x_0 = \left(\frac{1}{n^2}, \frac{1}{n^2}, \dots, \frac{1}{n^2}\right)^T.$$

Function 2. Trigonometric function

$$f_i(x) = 2(n + i(1 - \cos(x_i)) - \sin(x_i) - \sum_{k=1}^n \cos(x_k))(2 \sin(x_i) - \cos(x_i)), \quad i = 1, 2, \dots, n$$

$$\text{Initial guess: } x_0 = \left(\frac{101}{100n}, \frac{101}{100n}, \dots, \frac{101}{100n}\right)^T.$$

Function 3. Logarithmic function

$$f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad i = 1, 2, 3, \dots, n.$$

$$\text{Initial guess: } x_0 = (1, 1, \dots, 1)^T.$$

Function 4. Broyden Tridiagonal function [[52], pp. 471–472]

$$\begin{aligned} f_1(x) &= (3 - 0.5x_1)x_1 - 2x_2 + 1, \\ f_i(x) &= (3 - 0.5x_i)x_i - x_{i-1} + 2x_{i+1} + 1, \quad i = 2, 3, \dots, n-1, \\ f_n(x) &= (3 - 0.5x_n)x_n - x_{n-1} + 1. \end{aligned}$$

$$\text{Initial guess: } x_0 = (-1, -1, \dots, -1)^T.$$

Function 5. Strictly convex function 1 [[44], p. 29]

$$q(x) \text{ is the gradient of } h(x) = \sum_{i=1}^n (e^{x_i} - x_i).$$

$$f_i(x) = e^{x_i} - 1, \quad i = 1, 2, 3, \dots, n$$

$$\text{Initial guess: } x_0 = \left(\frac{1}{n}, \frac{2}{n}, \dots, 1\right)^T.$$

Function 6. Variable dimensioned function

$$\begin{aligned} f_i(x) &= x_i - 1, \quad i = 1, 2, 3, \dots, n-2, \\ f_{n-1}(x) &= \sum_{j=1}^{n-2} j(x_j - 1), \\ f_n(x) &= \left(\sum_{j=1}^{n-2} j(x_j - 1)\right)^2. \end{aligned}$$

$$\text{Initial guess: } x_0 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0\right)^T.$$

Function 7. Discrete boundary value problem [53].

$$\begin{aligned} f_1(x) &= 2x_1 + 0.5h^2(x_1 + h)^3 - x_2, \\ f_i(x) &= 2x_i + 0.5h^2(x_i + hi)^3 - x_{i-1} + x_{i+1}, \quad i = 2, 3, \dots, n-1 \\ f_n(x) &= 2x_n + 0.5h^2(x_n + hn)^3 - x_{n-1}, \\ h &= \frac{1}{n+1}. \end{aligned}$$

Initial guess: $x_0 = (h(h-1), h(2h-1), \dots, h(nh-1))^T$.

Function 8. Troesch problem [54]

$$\begin{aligned} f_1(x) &= 2x_1 + \varrho h^2 \sinh(\varrho x_1) - x_2 \\ f_i(x) &= 2x_i + \varrho h^2 \sinh(\varrho x_i) - x_{i-1} - x_{i+1}, \quad i = 2, 3, \dots, n-1 \\ f_n(x) &= 2x_n + \varrho h^2 \sinh(\varrho x_n) - x_{n-1}, \\ h &= \frac{1}{n+1}, \varrho = 10. \end{aligned}$$

Initial guess: $x_0 = (0, 0, \dots, 0)^T$.

By Tables 3 and 4, we see that Algorithm 5.1 and the PRP algorithm are effective for solving the above eight problems.

We use the tool of Dolan and Moré [35] to analyze the numerical performance of the two algorithms when NI, NG and cputime are considered, for which we generate three figures.

Fig 5 shows that the numerical performance of Algorithm 5.1 is slightly better than that of the PRP algorithm when NI is considered. It is easy to see that the numerical performance of

Fig 5. Performance profiles of the two algorithms (NI).

doi:10.1371/journal.pone.0140071.g005

Fig 6. Performance profiles of the two algorithms (NG).

doi:10.1371/journal.pone.0140071.g006

Fig 7. Performance profiles of the two algorithms (cputime).

doi:10.1371/journal.pone.0140071.g007

Algorithm 5.1 is better than that of the PRP algorithm from Figs 6 and 7 because the PRP algorithm requires a bigger horizontal axis when the problems are completely solved.

From the above two tables and three figures, we see that Algorithm 5.1 is effective and competitive for solving large-scale nonlinear equations.

Conclusion

(i) This paper provides the first new algorithm based on the first modified PRP conjugate gradient method in Sections 1–4. The β_k formula of the method includes the gradient value and function value. The global convergence of the algorithm is established under some suitable conditions. The trust region property and sufficient descent property of the method have been proved without the use of any line search method. For some test functions, the numerical results indicate that the first algorithm is effective and competitive for solving unconstrained optimization problems.

(ii) The second new algorithm based on the second modified PRP conjugate gradient method is presented in Sections 5–7. The new algorithm has global convergence under suitable conditions. The trust region property and the sufficient descent property of the method are proved without the use of any line search method. The numerical results of some tests function are demonstrated. The numerical results show that the second algorithm is very effective for solving large-scale nonlinear equations.

Acknowledgments

This work is supported by China NSF (Grant No. 11261006 and 11161003), NSFC No. 61232016, NSFC No. U1405254, the Guangxi Science Fund for Distinguished Young Scholars (No. 2015GXNSFGA139001) and PAPD issue of Jiangsu advantages discipline. The authors wish to thank the editor and the referees for their useful suggestions and comments which greatly improve this paper.

Author Contributions

Conceived and designed the experiments: GY XD WL. Performed the experiments: GY XD WL. Analyzed the data: GY XD WL XW ZC ZS. Contributed reagents/materials/analysis tools: GY XD WL XW ZC ZS. Wrote the paper: GY XD WL.

References

1. Gu B, Sheng V, Feasibility and finite convergence analysis for accurate on-line v-support vector learning, *IEEE Transactions on Neural Networks and Learning Systems*. 24 (2013) 1304–1315. doi: [10.1109/TNNLS.2013.2250300](https://doi.org/10.1109/TNNLS.2013.2250300)
2. Li J, Li X, Yang B, Sun X, Segmentation-based Image Copy-move Forgery Detection Scheme, *IEEE Transactions on Information Forensics and Security*. 10 (2015) 507–518. doi: [10.1109/TIFS.2014.2381872](https://doi.org/10.1109/TIFS.2014.2381872)
3. Wen X, Shao L, Fang W, Xue Y, Efficient Feature Selection and Classification for Vehicle Detection, *IEEE Transactions on Circuits and Systems for Video Technology*. 2015, doi: [10.1109/TCSVT.2014.2358031](https://doi.org/10.1109/TCSVT.2014.2358031)
4. Zhang H, Wu J, Nguyen TM, Sun M, Synthetic Aperture Radar Image Segmentation by Modified Student's t-Mixture Model, *IEEE Transaction on Geoscience and Remote Sensing*. 52 (2014) 4391–4403. doi: [10.1109/TGRS.2013.2281854](https://doi.org/10.1109/TGRS.2013.2281854)
5. Fu Z, Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing, *IEICE Transactions on Communications*. E98-B (2015) 190–200. doi: [10.1587/transcom.E98.B.190](https://doi.org/10.1587/transcom.E98.B.190)
6. Dai Y, Yuan Y, A nonlinear conjugate gradient with a strong global convergence properties, *SIAM J. Optim.* 10 (2000) 177–182. doi: [10.1137/S1052623497318992](https://doi.org/10.1137/S1052623497318992)

7. Fletcher R, Reeves C, Function minimization by conjugate gradients, *Comput. J.* 7 (1964) 149–154. doi: [10.1093/comjnl/7.2.149](https://doi.org/10.1093/comjnl/7.2.149)
8. Polak E, Ribière G, Note sur la convergence de directions conjugees, *Rev. Fran. Inf. Rech. Opérat.* 3 (1969) 35–43.
9. Polyak BT, The conjugate gradient method in extreme problems, *USSR Comput. Math. Math. Phys.* 9 (1969) 94–112. doi: [10.1016/0041-5553\(69\)90035-4](https://doi.org/10.1016/0041-5553(69)90035-4)
10. Hestenes MR, Stiefel EL, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand. Sect. B.* 49 (1952) 409–432. doi: [10.6028/res.049.044](https://doi.org/10.6028/res.049.044)
11. Liu Y, Storey C, Efficient generalized conjugate gradient algorithms part 1: theory. *J. Comput. Appl. Math.* 69 (1992) 17–41.
12. Fletcher R, *Practical method of optimization*, vol I: unconstrained optimization, 2nd edn. Wiley, New York, 1997.
13. Powell MJD, *Nonconvex minimization calculations and the conjugate gradient method*, *Lecture Notes in Mathematics*, vol. 1066, Springer, Berlin, 1984, pp. 122–141.
14. Gibert JC, Nocedal J, Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* 2 (1992) 21–42. doi: [10.1137/0802003](https://doi.org/10.1137/0802003)
15. Dai Y, *Analyses of conjugate gradient methods*, PH.D.thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences (in Chinese), 1997.
16. Ahmed T, Storey D, Efficient hybrid conjugate gradient techniques, *Journal of Optimization Theory and Applications.* 64 (1990) 379–394. doi: [10.1007/BF00939455](https://doi.org/10.1007/BF00939455)
17. Al-Baali A, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA Journal of Numerical Analysis.* 5 (1985) 121–124. doi: [10.1093/imanum/5.1.121](https://doi.org/10.1093/imanum/5.1.121)
18. Hu Y F, Storey C, Global convergence result for conjugate gradient methods, *Journal of Optimization Theory and Applications.* 71 (1991) 399–405. doi: [10.1007/BF00939927](https://doi.org/10.1007/BF00939927)
19. Yuan G, Wei Z, Zhao Q, A modified Polak-Ribière-Polyak conjugate gradient algorithm for large-scale optimization problems, *IIE Transactions.* 46 (2014), 397–413. doi: [10.1080/0740817X.2012.726757](https://doi.org/10.1080/0740817X.2012.726757)
20. Yuan G, Zhang M, A modified Hestenes-Stiefel conjugate gradient algorithm for large-scale optimization, *Numerical Functional Analysis and Optimization.* 34 (2013), 914–937. doi: [10.1080/01630563.2013.777350](https://doi.org/10.1080/01630563.2013.777350)
21. Yuan G, Lu X, Wei Z, A conjugate gradient method with descent direction for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 233 (2009), 519–530. doi: [10.1016/j.cam.2009.08.001](https://doi.org/10.1016/j.cam.2009.08.001)
22. Yuan G, Lu X, A modified PRP conjugate gradient method, *Annals of Operations Research.* 166 (2009), 73–90. doi: [10.1007/s10479-008-0420-4](https://doi.org/10.1007/s10479-008-0420-4)
23. Li X, Zhao X, A hybrid conjugate gradient method for optimization problems, *Natural Science.* 3(2011): 85–90. doi: [10.4236/ns.2011.31012](https://doi.org/10.4236/ns.2011.31012)
24. Yuan G, Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems, *Optimization Letters.* 3 (2009) 11–21. doi: [10.1007/s11590-008-0086-5](https://doi.org/10.1007/s11590-008-0086-5)
25. Huang H, Lin S, A modified Wei-Yao-Liu conjugate gradient method for unconstrained optimization, *Applied Mathematics and Computation.* 231 (2014) 179–186. doi: [10.1016/j.amc.2014.01.012](https://doi.org/10.1016/j.amc.2014.01.012)
26. Yu G, Zhao Y, Wei Z, A descent nonlinear conjugate gradient method for large-scale unconstrained optimization, *Applied mathematics and computation.* 187 (2) (2007) 636–643. doi: [10.1016/j.amc.2006.08.087](https://doi.org/10.1016/j.amc.2006.08.087)
27. Yao S, Wei Z, Huang H, A note about WYL's conjugate gradient method and its applications, *Applied Mathematics and computation.* 191 (2) (2007) 381–388. doi: [10.1016/j.amc.2007.02.094](https://doi.org/10.1016/j.amc.2007.02.094)
28. Hager WW, Zhang H, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM Journal on Optimization.* 16(1) (2005) 170–192. doi: [10.1137/030601880](https://doi.org/10.1137/030601880)
29. Wei Z, Yao S, Liu L, The convergence properties of some new conjugate gradient methods, *Applied Mathematics and Computation.* 183 (2006) 1341–1350. doi: [10.1016/j.amc.2006.05.150](https://doi.org/10.1016/j.amc.2006.05.150)
30. Zhang L, An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation, *Applied Mathematics and computation.* 215 (6) (2009) 2269–2274. doi: [10.1016/j.amc.2009.08.016](https://doi.org/10.1016/j.amc.2009.08.016)
31. Wei Z, Yu G, Yuan G, Lian Z, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, *Comput. Optim. Appl.* 29 (2004) 315–332. doi: [10.1023/B:COAP.0000044184.25410.39](https://doi.org/10.1023/B:COAP.0000044184.25410.39)
32. Yuan G, Wei Z, Convergence analysis of a modified BFGS method on convex minimizations, *Computational Optimization and Applications.* 47 (2) (2010) 237–255. doi: [10.1007/s10589-008-9219-0](https://doi.org/10.1007/s10589-008-9219-0)

33. Li M, Qu A, Some sufficient descent conjugate gradient methods and their global convergence, *Computational and Applied Mathematics*. 33 (2) (2014) 333–347. doi: [10.1007/s40314-013-0064-0](https://doi.org/10.1007/s40314-013-0064-0)
34. Zoutendijk Z, Nonlinear programming computational methods. In: Abadie J. (ed.) *Integer and nonlinear programming*, North-Holland, Amsterdam, 1970, 37–86.
35. Dolan ED, Moré JJ, Benchmarking optimization software with performance profiles, *Math. Program.* 91 (2002) 201–213. doi: [10.1007/s101070100263](https://doi.org/10.1007/s101070100263)
36. Buhmiller S, Krejic' N, Luzanin Z, Practical quasi-Newton algorithms for singular nonlinear systems, *Numer. Algorithms* 55 (2010) 481–502. doi: [10.1007/s11075-010-9367-z](https://doi.org/10.1007/s11075-010-9367-z)
37. Gu G, Li D, Qi L, Zhou S, Descent directions of quasi-Newton methods for symmetric nonlinear equations, *SIAMJ.Numer.Anal.* 40 (2002) 1763–1774. doi: [10.1137/S0036142901397423](https://doi.org/10.1137/S0036142901397423)
38. La Cruz W, Martínez. M, Raydan M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comp.* 75 (2006) 1429–1448. doi: [10.1090/S0025-5718-06-01840-0](https://doi.org/10.1090/S0025-5718-06-01840-0)
39. Kanzow C, Yamashita N, Fukushima M, Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties, *J. Comput. Appl. Math.* 172 (2004) 375–397. doi: [10.1016/j.cam.2004.02.013](https://doi.org/10.1016/j.cam.2004.02.013)
40. Zhang J, Wang Y, A new trust region method for nonlinear equations, *Math. Methods Oper. Res.* 58 (2003) 283–298. doi: [10.1007/s001860300302](https://doi.org/10.1007/s001860300302)
41. Grippo L, Sciandrone M, Nonmonotone derivative-free methods for nonlinear equations, *Comput. Optim. Appl.* 37 (2007) 297–328. doi: [10.1007/s10589-007-9028-x](https://doi.org/10.1007/s10589-007-9028-x)
42. Cheng W, A PRP type method for systems of monotone equations, *Math. Comput. Modelling.* 50 (2009) 15–20. doi: [10.1016/j.mcm.2009.04.007](https://doi.org/10.1016/j.mcm.2009.04.007)
43. La Cruz W, Raydan M, Nonmonotone spectral methods for large-scale nonlinear systems, *Optim. Methods Softw.* 18 (2003) 583–599. doi: [10.1080/10556780310001610493](https://doi.org/10.1080/10556780310001610493)
44. Raydan M, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.* 7 (1997) 26–33. doi: [10.1137/S1052623494266365](https://doi.org/10.1137/S1052623494266365)
45. Yu G, Guan L, Chen W, Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization, *Optim. Methods Softw.* 23 (2) (2008) 275–293. doi: [10.1080/10556780701661344](https://doi.org/10.1080/10556780701661344)
46. Yuan G, Wei Z, Lu S, Limited memory BFGS method with backtracking for symmetric nonlinear equations, *Math. Comput. Modelling.* 54 (2011) 367–377. doi: [10.1016/j.mcm.2011.02.021](https://doi.org/10.1016/j.mcm.2011.02.021)
47. Li D, Fukushima M, A global and superlinear convergent Gauss-Newton -based BFGS method for symmetric nonlinear equations, *SIAMJ.Numer.Anal.* 37 (1999) 152–172. doi: [10.1137/S0036142998335704](https://doi.org/10.1137/S0036142998335704)
48. Li Q, Li D, A class of derivative-free methods for large-scale nonlinear monotone equations, *IMA J. Numer. Anal.* 31 (2011) 1625–1635. doi: [10.1093/imanum/drq015](https://doi.org/10.1093/imanum/drq015)
49. Solodov MV, Svaiter BF, A hybrid projection-proximal point algorithm, *J. Convex Anal.* 6 (1999) 59–70.
50. Yuan G, Zhang M, A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations, *Journal of Computational and Applied Mathematics*. 286 (2015) 186–195. doi: [10.1016/j.cam.2015.03.014](https://doi.org/10.1016/j.cam.2015.03.014)
51. Solodov MV, Svaiter BF, A globally convergent inexact Newton method for systems of monotone equations, in: Fukushima M., Qi L. (Eds.), *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, Kluwer Academic Publishers. 1998, pp. 355–369.
52. Gomez-Ruggiero M, Martinez J, Moretti A, Comparing algorithms for solving sparse nonlinear systems of equations, *SIAM J. Sci. Comput.* 23 (1992) 459–483. doi: [10.1137/0913025](https://doi.org/10.1137/0913025)
53. Moré J, Garbow B, Hillström K, Testing unconstrained optimization software, *ACM Trans. Math. Softw.* 7 (1981) 17–41.
54. Roberts SM, Shipman JJ, On the closed form solution of Troesch's problem, *J. Comput. Phys.* 21 (1976) 291–304. doi: [10.1016/0021-9991\(76\)90026-7](https://doi.org/10.1016/0021-9991(76)90026-7)