A modified three-term PRP conjugate gradient algorithm for optimization models

Yanlin Wu*

*Correspondence: m1528083940@163.com
Department of Basic Teaching, Yango College, Fuzhou, Fujian 350015, P.R. China

Abstract

The nonlinear conjugate gradient (CG) algorithm is a very effective method for optimization, especially for large-scale problems, because of its low memory requirement and simplicity. Zhang et al. (IMA J. Numer. Anal. 26:629-649, 2006) firstly propose a three-term CG algorithm based on the well known Polak-Ribièr-Polyak (PRP) formula for unconstrained optimization, where their method has the sufficient descent property without any line search technique. They proved the global convergence of the Armijo line search but this fails for the Wolfe line search technique. Inspired by their method, we will make a further study and give a modified three-term PRP CG algorithm. The presented method possesses the following features: (1) The sufficient descent property also holds without any line search technique; (2) the trust region property of the search direction is automatically satisfied; (3) the step length is bounded from below; (4) the global convergence will be established under the Wolfe line search. Numerical results show that the new algorithm is more effective than that of the normal method.

MSC: 90C26

Keywords: conjugate gradient; sufficient descent; trust region

1 Introduction

We consider the optimization models defined by

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

(1.1)

where the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable. There exist many similar professional fields of science that can revert to the above optimization models (see, e.g., [2–21]). The CG method has the following iterative formula for (1.1):

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

(1.2)

where $x_k$ is the $k$th iterate point, the step length is $\alpha_k > 0$, and the search direction $d_k$ is designed by

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

(1.3)
where $g_k = \nabla f(x_k)$ is the gradient and $\beta_k \in \mathbb{R}$ is a scalar. At present, there are many well-known CG formulas (see [22–46]) and their applications (see, e.g., [47–50]), where one of the most efficient formulas is the PRP [34, 51] defined by

$$\beta^{\text{PRP}}_k = \frac{g^T_k \delta_k}{\|g_k\|^2}, \quad (1.4)$$

where $g_{k+1} = \nabla f(x_{k+1})$ is the gradient, $\delta_k = g_{k+1} - g_k$, and $\|\cdot\|$ is the Euclidian norm. The PRP method is very efficient as regards numerical performance, but it fails as regards the global convergence for the general functions under Wolfe line search technique and this is a still open problem; many scholars want to solve it. It is worth noting that a recent work of Yuan et al. [52] proved the global convergence of PRP method under a modified Wolfe line search technique for general functions. Al-Baali [53], Gilbert and Nocedal [54], Touati-Ahmed and Storey [55], and Hu and Storey [56] hinted that the sufficient descent property may be crucial for the global convergence of the conjugate gradient methods including the PRP method. Considering the above suggestions, Zhang, Zhou, and Li [1] firstly gave a three-term PRP formula

$$d_{k+1} = \begin{cases} 
-g_{k+1} + \frac{g^T_k \delta_k}{\|g_k\|^2}, \beta^{\text{PRP}}_k d_k - \vartheta_k \delta_k, & \text{if } k \geq 1, \\
-g_{k+1}, & \text{if } k = 0, 
\end{cases} \quad (1.5)$$

where $\vartheta_k = \frac{g^T_k d_k}{\|g_k\|^2}$. It is not difficult to deduce that $d^T_{k+1} g_{k+1} = -\|g_{k+1}\|^2$ holds for all $k$, which implies that the sufficient descent property is satisfied. Zhang et al. proved that the three-term PRP method has global convergence under Armijo line search technique for general functions but this fails for the Wolfe line search. The reason may be the trust region feature of the search direction that cannot be satisfied for this method. In order to overcome this drawback, we will propose a modified three-term PRP formula that will have not only the sufficient descent property but also the trust region feature.

In the next section, a modified three-term PRP formula is given and the new algorithm is stated. The sufficient descent property, the trust region feature, and the global convergence of the new method are established in Section 3. Numerical results are reported in the last section.

2 The modified PRP formula and algorithm

Motivated by the above observation, the modified three-term PRP formula is

$$d_{k+1} = \begin{cases} 
-g_{k+1} + \frac{g^T_k \delta_k d_k - \vartheta_k \delta_k}{\gamma_1 \|g_k\|^2 + \gamma_2 \|\delta_k\|^2 + \gamma_3 \|d_k\|^2}, \beta^{\text{PRP}}_k d_k - \vartheta_k \delta_k, & \text{if } k \geq 1, \\
-g_{k+1}, & \text{if } k = 0, 
\end{cases} \quad (2.1)$$

where $\gamma_1 > 0$, $\gamma_2 > 0$, and $\gamma_3 > 0$ are constants. It is easy to see that the difference between (1.5) and (2.1) is the denominator of the second and the third terms. This is a little change that will guarantee another good property for (2.1) and impel the global convergence for Wolfe conditions.
Algorithm 1 (New three-term PRP CG algorithm (NTT-PRP-CG-A))

Step 0: Initial given parameters: $x_1 \in \mathbb{R}^n$, $\gamma_1 > 0$, $\gamma_2 > 0$, $\gamma_3 > 0$, $0 < \delta < \sigma < 1$, $\varepsilon \in (0, 1)$. Let $d_1 = -g_1 = -\nabla f(x_1)$ and $k := 1$.

Step 1: $\|g_k\| \leq \varepsilon$, stop.

Step 2: Get stepsize $\alpha_k$ by the following Wolfe line search rules:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k,$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k.$$ (2.2)

Step 3: Let $x_{k+1} = x_k + \alpha_k d_k$. If the condition $\|g_{k+1}\| \leq \varepsilon$ holds, stop the program.

Step 4: Calculate the search direction $d_{k+1}$ by (2.1).

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

3 The sufficient descent property, the trust region feature, and the global convergence

It has been proved that, even for the function $f(x) = \lambda \|x\|^2$ ($\lambda > 0$ is a constant) and the strong Wolfe conditions, the PRP conjugate gradient method may not yield a descent direction for an unsuitable choice (see [24] for details). An interesting feature of the new three-term CG method is that the given search direction is sufficiently descent.

Lemma 3.1 The search direction $d_k$ is defined by (2.1) and it satisfies

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2$$ (3.1)

and

$$\|d_{k+1}\| \leq \gamma \|g_{k+1}\|$$ (3.2)

for all $k \geq 0$, where $\gamma > 0$ is a constant.

Proof For $k = 0$, it is easy to get $g_1^T d_1 = -g_1^T g_1 = -\|g_1\|^2$ and $\|d_1\| = \| - g_1\| = \|g_1\|$, so (3.1) is true and (3.2) holds with $\gamma = 1$.

If $k \geq 1$, by (2.1), we have

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + g_{k+1}^T \left[ \frac{g_{k+1}^T \delta_k d_k - d_{k+1}^T g_{k+1} \delta_k}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \delta_k + \gamma_3 \|d_k\| \|g_k\|} \right],$$

$$= -\|g_{k+1}\|^2 + \frac{g_{k+1}^T g_{k+1} \delta_k d_k - d_{k+1}^T g_{k+1} \delta_k}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \delta_k + \gamma_3 \|d_k\| \|g_k\|},$$

$$= -\|g_{k+1}\|^2.$$ (3.3)
Then (3.1) is satisfied. By (2.1) again, we obtain
\[\|d_{k+1}\| = \left\| \frac{g_{k+1}^T \delta_k d_k - d_{k+1}^T g_{k+1} \delta_k}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \|\delta_k\| + \gamma_3 \|d_k\| \|g_k\|} \right\| \]
\[\leq \|g_{k+1}\| + \left\| \frac{g_{k+1}^T \delta_k d_k - d_{k+1}^T g_{k+1} \delta_k}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \|\delta_k\| + \gamma_3 \|d_k\| \|g_k\|} \right\| \]
\[\leq \|g_{k+1}\| + \left\| \frac{\|\delta_k\| \|g_{k+1}\| \|d_k\| + \|d_k\| \|g_{k+1}\| \|\delta_k\|}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \|\delta_k\| + \gamma_3 \|d_k\| \|g_k\|} \right\| \]
\[\leq \|g_{k+1}\| + \left\| \frac{2\|\delta_k\| \|g_{k+1}\| \|d_k\|}{\gamma_2 \|d_k\| \|\delta_k\|} \right\| \]
\[= (1 + 2/\gamma_2)\|g_{k+1}\|, \quad (3.4)\]

where the last inequality follows from \(\frac{\|g_k\|^2 + \gamma_2 \|d_k\| \|\delta_k\| + \gamma_3 \|d_k\| \|g_k\|}{\gamma_1 \|g_k\|^2 + \gamma_2 \|d_k\| \|\delta_k\| + \gamma_3 \|d_k\| \|g_k\|} \leq \frac{1}{\gamma_2 \|d_k\| \|\delta_k\|} \). Thus (3.2) holds for all \(k \geq 0\) with \(\gamma = \max\{1, 1 + 2/\gamma_2\}\). The proof is complete. \(\square\)

Remark (1) Equation (3.1) is the sufficient descent property and the inequality (3.2) is the trust region feature. And these two relations are verifiable without any other conditions.

(2) Equations (3.1) and (2.2) imply that the sequence \(\{f(x_k)\}\) generated by Algorithm 1 is descent, namely \(f(x_k + \alpha_k d_k) \leq f(x_k)\) holds for all \(k\).

To establish the global convergence of Algorithm 1, the normal conditions are needed.

Assumption A

(i) The defined level set \(\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_1)\}\) is bounded with given point \(x_1\).
(ii) The function \(f\) has a lower bound and it is differentiable. The gradient \(g\) is Lipschitz continuous

\[\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n, \quad (3.5)\]

where \(L > 0\) a constant.

Lemma 3.2 Suppose that Assumption A holds and NTT-PRP-CG-A generates the sequence \(\{x_k, d_k, \alpha_k, g_k\}\). Then there exists a constant \(\beta > 0\) such that

\[\alpha_k \geq \beta, \quad \forall k \geq 1. \quad (3.6)\]

Proof Using (3.5) and (2.3) generate

\[\alpha_k L \geq (g_{k+1} - g_k)^T d_k \]
\[\geq -(1 - \sigma)\|g_k\|^2 d_k \]
\[= (1 - \sigma)\|g_k\|^2 \]

where the last equality follows from (3.1). By (3.2), we get

\[\alpha_k \geq \frac{1 - \sigma}{L \|d_k\|^2} \geq \frac{1 - \sigma}{L \gamma}. \]

Setting \(\beta \in (0, \frac{1 - \sigma}{L \gamma})\) completes the proof. \(\square\)
Remark. The above lemma shows that the steplength \( \alpha_k \) has a lower bound, which is helpful for the global convergence of Algorithm 1.

Theorem 3.1. Let the conditions of Lemma 3.2 hold and \( \{x_k, d_k, \alpha_k, g_k\} \) be generated by NTT-PRP-CG-A. Thus we get

\[
\lim_{k \to \infty} \|g_k\| = 0.
\]

Proof. By (2.2), (3.1), and (3.6), we have

\[
\delta \beta \|g_k\|^2 \leq \delta \alpha_k \|g_k\|^2 \leq f(x_k) - f(x_k + \alpha_k d_k).
\]

Summing the above inequality from \( k = 1 \) to \( \infty \), we have

\[
\sum_{k=1}^{\infty} \delta \beta \|g_k\|^2 \leq f(x_1) - f_\infty \leq \infty,
\]

which means that

\[
\|g_k\| \to 0, \quad k \to \infty.
\]

The proof is complete. \( \Box \)

4 Numerical results and discussion

This section will report the numerical experiment of the NTT-PRP-CG-A and the algorithm of Zhang et al. [1] (called Norm-PRP-A), where the Norm-PRP-A is the Step 4 of Algorithm 1 that is replaced by: Calculate the search direction \( d_{k+1} \) by (1.5). Since the method is based on the search direction (1.5), we only compare the numerical results between the new algorithm and the Norm-PRP-A. The codes of these two algorithms are written by Matlab and the problems are chosen from [57, 58] with given initial points. The tested problems are listed in Table 1. The parameters are \( \gamma_1 = 2 \), \( \gamma_2 = 5 \), \( \gamma_3 = 3 \), \( \delta = 0.01 \), \( \sigma = 0.86 \). The program uses the Himmelblau rule: Set \( S_1 = \frac{f(x_k) - f(x_{k+1})}{\|f(x_k)\|} \) if \( |f(x_k)| > \tau_1 \), otherwise set \( S_1 = |f(x_k) - f(x_{k+1})| \). The program stops if \( \|g(x)\| < \epsilon \) or \( S_1 < \tau_2 \) hold, where we choose

| No. | Problems                              | \( x_0 \)                        |
|-----|---------------------------------------|----------------------------------|
| 1   | Extended Freudenstein and Roth function | \([0.5, -2, \ldots, 0.5, -2]\)  |
| 2   | Extended trigonometric function        | \([0.2, 0.2, \ldots, 0.2]\)     |
| 3   | Extended Rosenbrock function          | \([-1.2, 1, \ldots, -1.2, 1]\)  |
| 4   | Extended White and Holst function     | \([-1.2, 1, \ldots, -1.2, 1]\)  |
| 5   | Extended Beale function               | \([1, 0.8, \ldots, 1/0.8]\)     |
| 6   | Extended penalty function             | \([1/1, 1/2, \ldots, 1/n]\)     |
| 7   | Perturbed quadratic function          | \([0.5, 0.5, \ldots, 0.5]\)     |
| 8   | Raydan 1 function                     | \([1, \ldots, 1]\)             |
| 9   | Raydan 2 function                     | \([1, \ldots, 1]\)             |
| 10  | Diagonal 1 function                   | \([1/n, 1/n, \ldots, 1/n]\)    |
| 11  | Diagonal 2 function                   | \([1/1, 1/2, \ldots, 1/n]\)    |
| 12  | Diagonal 3 function                   | \([1, 1, \ldots, 1]\)          |
| No. | Problems                                      | $x_0$                          |
|-----|----------------------------------------------|--------------------------------|
| 13  | Hager function                              | [1, 1, ..., 1]                 |
| 14  | Generalized tridiagonal 1 function          | [2, 2, ..., 2]                 |
| 15  | Extended tridiagonal 1 function             | [2, 2, ..., 2]                 |
| 16  | Extended three exponential terms function   | [0, 1, 0.1, ..., 0.1]          |
| 17  | Generalized tridiagonal 2 function          | [−1, −1, ..., −1, −1]          |
| 18  | Diagonal 4 function                         | [1, 1, 1, 1]                   |
| 19  | Diagonal 5 function                         | [1, 1, 1, 1, 1]                |
| 20  | Extended Himmelblau function                | [1, 1, 1]                      |
| 21  | Generalized PSC1 function                   | [3, 0.1, ..., 3, 0.1]          |
| 22  | Extended PSC1 function                      | [3, 0.1, ..., 3, 0.1]          |
| 23  | Extended Powell function                     | [3, −1, 0, 1, ...]            |
| 24  | Extended block diagonal BD1 function        | [0, 1, 0.1, ..., 0.1]          |
| 25  | Extended Maratos function                   | [1, 1, 0.1, ..., 1, 1, 0.1]   |
| 26  | Extended Cliff function                     | [0, −1, ..., 0, −1]           |
| 27  | Quadratic diagonal perturbed function       | [0.5, 0.5, ..., 0.5]           |
| 28  | Extended Wood function                      | [−3, −1, −3, −1, ..., −3, −1]  |
| 29  | Extended Hiebert function                   | [0, 0, ..., 0]                 |
| 30  | Quadratic QF1 function                      | [1, 1, ..., 1]                 |
| 31  | Extended quadratic penalty QP1 function     | [1, 1, ..., 1]                 |
| 32  | Extended quadratic penalty QP2 function     | [1, 1, ..., 1]                 |
| 33  | Quadratic QF2 function                      | [0.5, 0.5, ..., 0.5]           |
| 34  | Extended EP1 function                       | [1.5, 1.5, ..., 1.5]           |
| 35  | Extended tridiagonal-2 function             | [1, 1, 1]                      |
| 36  | BDORTIC function (CUTE)                     | [1, 1, 1]                      |
| 37  | TRIDIA function (CUTE)                      | [1, 1, 1]                      |
| 38  | ARWHEAD function (CUTE)                     | [1, 1, 1]                      |
| 39  | NONDIA (Shanno-78) function (CUTE)          | [−1, −1, ..., −1, −1]          |
| 40  | NONDIQUR function (CUTE)                    | [1, −1, 1, −1, ..., 1, −1]    |
| 41  | DQORTIC function (CUTEr)                    | [3, 3, 3, 3]                   |
| 42  | EG2 function (CUTE)                         | [1, 1, 1]                      |
| 43  | DIXMAANA function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 44  | DIXMAANB function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 45  | DIXMAANC function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 46  | DIXMAANE function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 47  | Partial perturbed quadratic function        | [0.5, 0.5, ..., 0.5]           |
| 48  | Broyden tridiagonal function                | [−1, −1, ..., −1]              |
| 49  | Almost perturbed quadratic function         | [0.5, 0.5, ..., 0.5]           |
| 50  | Tridiagonal perturbed quadratic function    | [0.5, 0.5, ..., 0.5]           |
| 51  | EDENSCH function (CUTE)                     | [0, 0, ..., 0]                 |
| 52  | VARDIM function (CUTE)                      | [1 − 1/n, 1 − 2/n, ..., 1 − n/n]|
| 53  | STAIRCASE S1 function                       | [1, 1, 1]                      |
| 54  | LIARWHD function (CUTEr)                    | [4, 4, 4, 4]                   |
| 55  | DIAGONAL 6 function                         | [1, 1, 1]                      |
| 56  | DIXON3DQ function (CUTE)                    | [−1, −1, ..., −1]              |
| 57  | DIXMAANF function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 58  | DIXMAANG function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 59  | DIXMAANH function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 60  | DIXMAANl function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 61  | DIXMAANJ function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 62  | DIXMAANK function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 63  | DIXMAANL function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 64  | DIXMAAND function (CUTE)                    | [2, 2, 2, ..., 2]              |
| 65  | ENGVAL1 function (CUTE)                     | [2, 2, 2, ..., 2]              |
| 66  | FLETCHCR function (CUTE)                    | [0, 0, ..., 0]                 |
| 67  | COSINE function (CUTE)                      | [1, 1, 1]                      |
| 68  | Extended DENSCHNB function (CUTE)           | [1, 1, 1]                      |
| 69  | DENSCHNF function (CUTEr)                   | [2, 0, 2, 0, ..., 2, 0]        |
| 70  | SINOQUAD function (CUTE)                    | [0, 1, 0.1, ..., 0.1]          |
| 71  | BIGGSB1 function (CUTE)                     | [0, 0, ..., 0]                 |
| 72  | Partial perturbed quadratic PPQ2 function   | [0.5, 0.5, ..., 0.5]           |
| 73  | Scaled quadratic SQ1 function               | [1, 2, ..., n]                 |
| 74  | Scaled quadratic SQ2 function               | [1, 2, ..., n]                 |
\[ \epsilon = 10^{-6} \text{ and } \tau_1 = \tau_2 = 10^{-5}. \]

For the choice of the stepsize \( \alpha_k \) in (2.2) and (2.3), the largest cycle number is 10 and the current stepsize is accepted. The dimensions of the test problems accord to large-scale variables with 3,000, 12,000, and 30,000. The runtime environment is MATLAB R2010b and run on a PC with CPU Intel Pentium(R) Dual-Core CPU at 3.20 GHz, 2.00 GB of RAM, and the Windows 7 operating system.

Table 2 report the test numerical results of the NTT-PRP-CG-A and the Norm-PRP-A, and we note:

No. the test problems number. Dimension: the variables number.
Ni: the iteration number. Nfg: the function and the gradient value number. CPU time: the CPU time of operating system in seconds.

| No. | Dimension | NTT-PRP-CG-A | Norm-PRP-A |
|-----|-----------|--------------|------------|
|     | Ni     | Nfg       | CPU time  | Ni     | Nfg       | CPU time  |
| 1   | 3,000  | 15        | 43        | 0.468003 | 31        | 92        | 0.546004 |
| 12,000 | 15    | 43        | 0.842405  | 56       | 158       | 1.778411  |
| 30,000 | 15    | 43        | 1.482009  | 36       | 113       | 2.730018  |
| 2   | 3,000  | 57        | 131       | 0.374402 | 55        | 126       | 0.374402  |
| 12,000 | 63    | 144       | 1.138807  | 62       | 142       | 0.920406  |
| 30,000 | 66    | 152       | 3.08882   | 66       | 152       | 2.511616  |
| 3   | 3,000  | 54        | 186       | 0.124801 | 117       | 375       | 0.202801  |
| 12,000 | 67    | 233       | 0.234001  | 144      | 479       | 0.514803  |
| 30,000 | 73    | 253       | 0.530403  | 159      | 522       | 1.624216  |
| 4   | 3,000  | 59        | 198       | 0.296402 | 207       | 595       | 0.936006  |
| 12,000 | 34    | 139       | 0.733205  | 264      | 801       | 4.305638  |
| 30,000 | 74    | 256       | 4.118426  | 228      | 618       | 8.907657  |
| 5   | 3,000  | 23        | 68        | 0.093601 | 39        | 106       | 0.124801  |
| 12,000 | 23    | 69        | 0.265202  | 39       | 109       | 0.390003  |
| 30,000 | 21    | 64        | 0.826805  | 47       | 135       | 1.279208  |
| 6   | 3,000  | 80        | 185       | 0.124801 | 80        | 185       | 0.093601  |
| 12,000 | 103   | 232       | 0.405603  | 103      | 232       | 0.343202  |
| 30,000 | 102   | 235       | 1.216808  | 102      | 235       | 0.998406  |
| 7   | 3,000  | 1,000     | 2,002     | 3.16682  | 835       | 2,257     | 2.808018  |
| 12,000 | 1,000 | 2,002     | 9.781263  | 1,000    | 2,779     | 9.734462  |
| 30,000 | 1,000 | 2,002     | 9.781263  | 1,000    | 2,779     | 9.734462  |
| 8   | 3,000  | 21        | 47        | 0.0468   | 19        | 46        | 0.0312    |
| 12,000 | 20    | 44        | 0.093601  | 19       | 46        | 0.093601  |
| 30,000 | 20    | 44        | 0.296402  | 19       | 46        | 0.265202  |
| 9   | 3,000  | 12        | 26        | 0.0312   | 12        | 26        | 0.0312    |
| 12,000 | 12    | 26        | 0.0468   | 12       | 26        | 0.0624    |
| 30,000 | 12    | 26        | 0.202801  | 12       | 26        | 0.156001  |
| 10  | 3,000  | 2         | 13        | 0.0312   | 2         | 13        | 0.0312    |
| 12,000 | 2     | 13        | 0.124801  | 2        | 13        | 0.093601  |
| 30,000 | 2     | 13        | 0.312002  | 2        | 13        | 0.280802  |
| 11  | 3,000  | 81        | 194       | 0.171601 | 24        | 101       | 0.0624    |
| 12,000 | 91    | 247       | 0.764405  | 15       | 59        | 0.202801  |
| 30,000 | 11    | 35        | 0.436803  | 13       | 50        | 0.280802  |
| 12  | 3,000  | 17        | 36        | 0.0468   | 14        | 33        | 0.0624    |
| 12,000 | 19    | 40        | 0.171601  | 14       | 33        | 0.124801  |
| 30,000 | 19    | 40        | 0.499203  | 14       | 33        | 0.343202  |
| 13  | 3,000  | 23        | 86        | 0.093601 | 22        | 84        | 0.078     |
| 12,000 | 42    | 111       | 0.452403  | 42       | 111       | 0.468003  |
| 30,000 | 2     | 13        | 0.358802  | 2        | 13        | 0.327602  |
| No. | Dimension | NTT-PRP-CG-A | Norm-PRP-A |
|-----|-----------|-------------|------------|
|     | Ni Nfg CPU time | Ni Nfg CPU time |             |
| 14  | 3,000 | 6 15 0.717605 6 15 0.733205 |
| 12,000 | 6 15 7.004445 5 13 5.709637 |
| 30,000 | 3 8 14.258491 3 8 13.587687 |
| 15  | 3,000 | 38 85 1.794011 66 176 3.04202 |
| 12,000 | 41 102 17.924515 60 169 28.09578 |
| 30,000 | 44 114 75.395283 68 194 120.245571 |
| 16  | 3,000 | 20 42 0.0624 20 42 0 |
| 12,000 | 24 50 0.171601 24 50 0.156001 |
| 30,000 | 24 50 0.483603 24 50 0.436803 |
| 17  | 3,000 | 24 55 0.156001 31 71 0.218401 |
| 12,000 | 33 73 0.764405 29 74 0.717605 |
| 30,000 | 48 103 3.042019 30 81 1.996813 |
| 18  | 3,000 | 3 10 0.0156 13 43 0.0312 |
| 12,000 | 3 10 0.0312 13 43 0.0156 |
| 30,000 | 3 10 0.0312 14 47 0.124801 |
| 19  | 3,000 | 3 9 0 3 9 0 |
| 12,000 | 3 9 0.0468 3 9 0.0312 |
| 30,000 | 3 9 0.124801 3 9 0.124801 |
| 20  | 3,000 | 33 82 0.0312 26 74 0.0312 |
| 12,000 | 11 61 0.0624 5 35 0.0312 |
| 30,000 | 5 35 0.093601 20 67 0.218401 |
| 21  | 3,000 | 25 59 0.093601 27 63 0.0624 |
| 12,000 | 27 63 0.249602 26 60 0.187201 |
| 30,000 | 25 58 0.530403 27 63 0.530403 |
| 22  | 3,000 | 6 31 0.0312 7 42 0 |
| 12,000 | 6 31 0.0624 5 21 0.0624 |
| 30,000 | 6 31 0.218401 5 21 0.124801 |
| 23  | 3,000 | 134 383 0.670804 334 986 1.52881 |
| 12,000 | 147 416 2.652017 452 1309 7.73765 |
| 30,000 | 114 330 5.304034 291 854 12.776482 |
| 24  | 3,000 | 28 90 0.0624 50 126 0.109201 |
| 12,000 | 31 108 0.249602 60 146 0.405603 |
| 30,000 | 28 97 0.686404 67 160 1.170007 |
| 25  | 3,000 | 28 56 0.0312 28 56 0.0312 |
| 12,000 | 7 16 0.0156 231 774 0.748805 |
| 30,000 | 7 16 0.0312 213 774 0.208013 |
| 26  | 3,000 | 65 152 0.124801 65 152 0.124801 |
| 12,000 | 72 166 0.514803 72 166 0.468003 |
| 30,000 | 79 180 1.51321 79 180 1.341609 |
| 27  | 3,000 | 31 94 0.0624 104 327 0.156001 |
| 12,000 | 43 137 0.187201 202 655 0.639604 |
| 30,000 | 104 329 1.154407 384 1231 4.024826 |
| 28  | 3,000 | 40 124 0.0468 31 76 0.0312 |
| 12,000 | 31 91 0.124801 38 95 0.124801 |
| 30,000 | 40 107 0.546003 32 78 0.265202 |
| 29  | 3,000 | 4 19 0.0312 100 287 0.124801 |
| 12,000 | 4 19 0.0156 84 240 0.312002 |
| 30,000 | 4 19 0.093601 93 264 0.842405 |
| 30  | 3,000 | 1,000 2,002 0.842405 446 1205 0.436803 |
| 12,000 | 1,000 2,002 2.636417 754 2010 2.074813 |
| 30,000 | 1,000 2,002 8.330453 1,000 2,721 8.065252 |
| 31  | 3,000 | 29 66 0.0468 29 66 0.0624 |
| 12,000 | 34 78 0.156001 34 78 0.156001 |
| 30,000 | 34 78 0.421203 34 78 0.452403 |
Table 2 (Continued)

| No. | Dimension NTT-PRP-CG-A |     | | Norm-PRP-A |     |
|-----|------------------------|-----|-----|-------------|-----|
|     | Ni | Nfg | CPU time | Ni | Nfg | CPU time |
| 32  | 3,000 | 48 | 100 | 0.093601 | 48 | 100 | 0.093601 |
|     | 12,000 | 37 | 80 | 0.280802 | 37 | 80 | 0.234001 |
|     | 30,000 | 36 | 80 | 0.780005 | 36 | 80 | 0.670804 |
| 33  | 3,000 | 3 | 7 | 0 | 3 | 7 | 0 |
|     | 12,000 | 2 | 5 | 0 | 2 | 5 | 0.0312 |
|     | 30,000 | 2 | 5 | 0.0312 | 2 | 5 | 0 |
| 34  | 3,000 | 4 | 8 | 0.0312 | 4 | 8 | 0.0312 |
|     | 12,000 | 7 | 14 | 0.0624 | 7 | 14 | 0.0312 |
|     | 30,000 | 10 | 20 | 0.156001 | 10 | 20 | 0.124801 |
| 35  | 3,000 | 12 | 24 | 0.0312 | 12 | 24 | 0 |
|     | 12,000 | 21 | 42 | 0.093601 | 21 | 42 | 0.093601 |
|     | 30,000 | 4 | 10 | 0.093601 | 4 | 10 | 0.0312 |
| 36  | 3,000 | 14 | 48 | 1.138807 | 45 | 148 | 3.244821 |
|     | 12,000 | 8 | 28 | 6.970844 | 120 | 369 | 95.821414 |
|     | 30,000 | 17 | 55 | 55.427155 | 162 | 483 | 488.922734 |
| 37  | 3,000 | 776 | 1,559 | 0.733205 | 1,000 | 2,688 | 1.107607 |
|     | 12,000 | 1,000 | 2,006 | 3.322821 | 1,000 | 2,733 | 3.556823 |
|     | 30,000 | 1,000 | 2,011 | 9.828063 | 506 | 1,378 | 4.960832 |
| 38  | 3,000 | 9 | 30 | 0.0312 | 27 | 81 | 0.0312 |
|     | 12,000 | 10 | 32 | 0.0468 | 21 | 60 | 0.140401 |
|     | 30,000 | 11 | 34 | 0.140401 | 24 | 64 | 0.312002 |
| 39  | 3,000 | 26 | 52 | 0.0624 | 26 | 52 | 0 |
|     | 12,000 | 29 | 58 | 0.093601 | 29 | 58 | 0.093601 |
|     | 30,000 | 23 | 46 | 0.187201 | 23 | 46 | 0.171601 |
| 40  | 3,000 | 554 | 1,332 | 5.881238 | 1,000 | 2,856 | 11.013671 |
|     | 12,000 | 1,000 | 2,228 | 39.733455 | 1,000 | 2,892 | 43.352678 |
|     | 30,000 | 1,000 | 2,247 | 100.745446 | 1,000 | 2,866 | 108.16694 |
| 41  | 3,000 | 27 | 68 | 0.078 | 49 | 133 | 0.0312 |
|     | 12,000 | 28 | 69 | 0.093601 | 50 | 136 | 0.124801 |
|     | 30,000 | 37 | 91 | 0.390002 | 39 | 101 | 0.374402 |
| 42  | 3,000 | 6 | 24 | 0.0312 | 6 | 24 | 0 |
|     | 12,000 | 6 | 24 | 0.0624 | 6 | 24 | 0.0624 |
|     | 30,000 | 6 | 24 | 0.187201 | 6 | 24 | 0.156001 |
| 43  | 3,000 | 28 | 60 | 0.202801 | 28 | 60 | 0.218401 |
|     | 12,000 | 30 | 64 | 0.936006 | 30 | 64 | 0.858005 |
|     | 30,000 | 32 | 68 | 2.527216 | 31 | 66 | 2.230814 |
| 44  | 3,000 | 46 | 96 | 0.358802 | 46 | 96 | 0.296402 |
|     | 12,000 | 49 | 102 | 1.51321 | 49 | 102 | 1.400409 |
|     | 30,000 | 52 | 108 | 4.024826 | 52 | 108 | 3.728424 |
| 45  | 3,000 | 19 | 44 | 0.202801 | 19 | 44 | 0.124801 |
|     | 12,000 | 20 | 46 | 0.608404 | 20 | 46 | 0.577204 |
|     | 30,000 | 20 | 46 | 1.54441 | 20 | 46 | 1.48201 |
| 46  | 3,000 | 117 | 244 | 0.920406 | 108 | 296 | 0.967206 |
|     | 12,000 | 165 | 340 | 5.116833 | 120 | 326 | 4.009226 |
|     | 30,000 | 195 | 400 | 15.678101 | 126 | 341 | 10.576868 |
| 47  | 3,000 | 27 | 66 | 8.299253 | 44 | 102 | 12.963683 |
|     | 12,000 | 31 | 87 | 93.741001 | 49 | 141 | 150.150963 |
|     | 30,000 | 69 | 182 | 1,163.84546 | 85 | 256 | 1,490.683156 |
| 48  | 3,000 | 32 | 74 | 1.762811 | 27 | 63 | 1.310408 |
|     | 12,000 | 50 | 103 | 30.154993 | 29 | 74 | 19.546925 |
|     | 30,000 | 42 | 100 | 112.726323 | 37 | 87 | 94.209004 |
| 49  | 3,000 | 1,000 | 2,002 | 0.858005 | 575 | 1,593 | 0.577204 |
|     | 12,000 | 1,000 | 2,002 | 2.792418 | 885 | 2,377 | 2.527216 |
|     | 30,000 | 1,000 | 2,002 | 9.484861 | 1,000 | 2,738 | 8.143252 |
| No. | Dimension | NTT-PRP-CG-A | Norm-PRP-A |
|-----|-----------|--------------|------------|
|     |           | Ni | Nfg | CPUtime | Ni | Nfg | CPUtime |
| 50  | 3,000     | 1,000 | 2,002 | 57.236767 | 370 | 998 | 23.727752 |
| 12,000 | 3,000 | 2,002 | 617.73276 | 920 | 2,495 | 676.23313 |
| 30,000 | 1,000 | 2,002 | 2,467.96702 | 1,000 | 2,720 | 2,856.471911 |
| 51  | 3,000     | 23 | 48 | 0.124801 | 23 | 48 | 0.140401 |
| 12,000 | 23 | 48 | 0.811205 | 23 | 48 | 0.452403 |
| 30,000 | 121 | 276 | 2,090413 | 128 | 316 | 1,684811 |
| 52  | 3,000     | 138 | 316 | 2,090413 | 138 | 316 | 1,684811 |
| 12,000 | 150 | 344 | 4,66443 | 150 | 344 | 4,61763 |
| 30,000 | 1,000 | 2,009 | 3.759624 | 1,000 | 2,661 | 3.369622 |
| 53  | 3,000     | 23 | 48 | 0.0624 | 23 | 48 | 0.109201 |
| 12,000 | 23 | 48 | 0.811205 | 23 | 48 | 0.452403 |
| 30,000 | 23 | 48 | 1.154407 | 23 | 48 | 1.216808 |
| 54  | 3,000     | 121 | 276 | 0.436803 | 121 | 276 | 0.374402 |
| 12,000 | 138 | 316 | 7.488048 | 138 | 316 | 7.441248 |
| 30,000 | 150 | 344 | 4.66443 | 150 | 344 | 4.61763 |
| 55  | 3,000     | 1,000 | 2,009 | 0.998406 | 1,000 | 2,706 | 1.170008 |
| 12,000 | 1,000 | 2,009 | 3.759624 | 1,000 | 2,661 | 3.369622 |
| 30,000 | 1,000 | 2,009 | 8.502054 | 1,000 | 2,781 | 9.594061 |
| 56  | 3,000     | 430 | 886 | 0.358802 | 507 | 1,397 | 0.608404 |
| 12,000 | 430 | 886 | 1.450809 | 613 | 1,667 | 2.043613 |
| 30,000 | 430 | 886 | 3.541223 | 491 | 1,337 | 4.492829 |
| 57  | 3,000     | 145 | 296 | 1.154407 | 55 | 132 | 0.468003 |
| 12,000 | 145 | 296 | 7.75325 | 69 | 179 | 2.246414 |
| 30,000 | 265 | 536 | 19.500125 | 77 | 196 | 6.27124 |
| 58  | 3,000     | 107 | 223 | 0.873606 | 107 | 223 | 0.873606 |
| 12,000 | 107 | 223 | 3.931225 | 91 | 243 | 3.07322 |
| 30,000 | 142 | 293 | 10.514467 | 98 | 261 | 8.205653 |
| 59  | 3,000     | 77 | 166 | 0.639604 | 52 | 137 | 0.405603 |
| 12,000 | 107 | 226 | 4.08429 | 60 | 152 | 1.934412 |
| 30,000 | 94 | 203 | 7.082445 | 72 | 181 | 5.803237 |
| 60  | 3,000     | 488 | 983 | 3.978026 | 111 | 303 | 0.967206 |
| 12,000 | 175 | 360 | 5.522435 | 106 | 293 | 3.650423 |
| 30,000 | 194 | 398 | 14.47693 | 140 | 377 | 11.856076 |
| 61  | 3,000     | 145 | 296 | 1.185608 | 56 | 142 | 0.468003 |
| 12,000 | 206 | 418 | 6.692443 | 70 | 179 | 2.277615 |
| 30,000 | 264 | 534 | 19.390924 | 92 | 247 | 7.75325 |
| 62  | 3,000     | 153 | 314 | 1.232408 | 63 | 163 | 0.717605 |
| 12,000 | 239 | 486 | 7.332047 | 86 | 214 | 2.761218 |
| 30,000 | 313 | 634 | 23.166148 | 96 | 261 | 8.127652 |
| 63  | 3,000     | 209 | 430 | 1.934412 | 138 | 378 | 1.388409 |
| 12,000 | 1,000 | 2,009 | 3.779042 | 164 | 448 | 6.489642 |
| 30,000 | 1,000 | 2,009 | 8.7220159 | 191 | 521 | 18.532919 |
| 64  | 3,000     | 29 | 64 | 0.265202 | 29 | 64 | 0.218401 |
| 12,000 | 31 | 68 | 1.045207 | 31 | 68 | 0.936006 |
| 30,000 | 32 | 70 | 2.340015 | 32 | 70 | 2.324415 |
| 65  | 3,000     | 22 | 51 | 1.903212 | 19 | 45 | 1.59121 |
| 12,000 | 17 | 38 | 14.586094 | 17 | 38 | 14.258491 |
| 30,000 | 17 | 38 | 61.167992 | 17 | 38 | 59.420781 |
| 66  | 3,000     | 1,000 | 2,003 | 57.985572 | 733 | 2,293 | 50.684725 |
| 12,000 | 1,000 | 2,003 | 618.637566 | 214 | 671 | 171.757101 |
| 30,000 | 4 | 11 | 10.374067 | 58 | 157 | 163.879051 |
| 67  | 3,000     | 6 | 37 | 0.0312 | 9 | 59 | 0.0312 |
| 12,000 | 10 | 63 | 0.499203 | 48 | 231 | 0.577204 |
| 30,000 | 5 | 27 | 0.124801 | 10 | 54 | 0.296402 |
Table 2 (Continued)

| No. | Dimension | NTT-PRP-CG-A | Norm-PRP-A |
|-----|-----------|--------------|------------|
|     |           | Ni | Nfg | CPU time | Ni | Nfg | CPU time |
| 68  | 3,000     | 35 | 72  | 0.0312   | 35 | 72  | 0.0312   |
|     | 12,000    | 38 | 78  | 0.124801 | 38 | 78  | 0.109201 |
|     | 30,000    | 39 | 80  | 0.343202 | 39 | 80  | 0.374402 |
| 69  | 3,000     | 27 | 58  | 0.0312   | 30 | 64  | 0.0624   |
|     | 12,000    | 28 | 60  | 0.140401 | 32 | 68  | 0.187201 |
|     | 30,000    | 29 | 62  | 0.421203 | 33 | 70  | 0.468003 |
| 70  | 3,000     | 25 | 82  | 1.950013 | 129| 386 | 8.876457 |
|     | 12,000    | 52 | 184 | 46.8471  | 143| 479 | 119.621567 |
|     | 30,000    | 13 | 62  | 52.790738| 193| 598 | 597.967433 |
| 71  | 3,000     | 1,000| 2,004| 0.889206 | 449| 1,247| 0.468003 |
|     | 12,000    | 1,000| 2,004| 4.196427 | 661| 1,779| 2.106014 |
|     | 30,000    | 706| 2,011| 228.837867| 1,000| 2,845| 323.40567 |
| 72  | 3,000     | 569| 1,589| 1,742.46877| 785| 2,234| 2,412.04662 |
|     | 12,000    | 229| 654 | 3,931.381201| 1,000| 2,813| 17,084.27791 |
| 73  | 3,000     | 1,000| 2,002| 0.936006 | 490| 1,307| 0.421203 |
|     | 12,000    | 1,000| 2,002| 3.291621 | 900| 2,460| 2.605217 |
|     | 30,000    | 785| 2,234| 7.566048 | 1,000| 2,735| 7.940451 |
| 74  | 3,000     | 1,000| 2,002| 0.873606 | 398| 1,061| 0.374402 |
|     | 12,000    | 1,000| 2,002| 4.399228 | 795| 2,120| 2.262015 |
|     | 30,000    | 1,000| 2,002| 7.519248 | 1,000| 2,682| 7.86245 |

A new tool was given by Dolan and Moré [59] to analyze the performance of the algorithms. Figures 1-3 show that the efficiency of the NTT-PRP-CG-A and the Norm-PRP-A relate to Ni, Nfg, and CPU time, respectively. It is easy to see that these two algorithms are effective for those problems and the given three-term PRP conjugate gradient method is more effective than that of the normal three-term PRP conjugate gradient method. Moreover, the NTT-PRP-CG-A has good robustness. Overall, the presented algorithm has some potential property both in theory and numerical experiment, which is noticeable.
5 Conclusions
In this paper, based on the PRP formula for unconstrained optimization, a modified three-term PRP CG algorithm was presented. The proposed method possesses sufficient descent property also holds without any line search technique, and we have automatically the trust region property of the search direction. Under the Wolfe line search, the global convergence was proven. Numerical results showed that the new algorithm is more effective compared with the normal method.

Competing interests
The author declares that they have no competing interests.
Author's contributions
Only the author contributed in writing this paper.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 March 2017 Accepted: 20 April 2017 Published online: 03 May 2017

References
1. Zhang, L, Zhou, W, Li, D: A descent modified Polak-Ribiére-Polyak conjugate method and its global convergence. IMJ Numer. Anal. 26, 629-649 (2006)
2. Fu, Z, Wu, X, Guan, C, Sun, X, Ren, K: Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans. Inf. Forensics Secur. (2016), doi:10.1109/TIFS.2016.2596138
3. Gu, B, Sheng, VS, Tay, KY, Romano, W, Li, S: Incremental support vector learning for ordinal regression. IEEE Trans. Neural Netw. Learn. Syst. 26, 1403-1416 (2015)
4. Gu, B, Sun, X, Sheng, VS: Structural minimax probability machine, IEEE Trans. Neural Netw. Learn. Syst. (2016). doi:10.1109/TNNLS.2016.2544779
5. Li, J, Li, X, Yang, B, Sun, X: Segmentation-based image copy-move forgery detection scheme. IEEE Trans. Inf. Forensics Secur. 10, 507-518 (2015)
6. Pan, Z, Lei, J, Zhang, Y, Sun, X, Kwong, S: Fast motion estimation based on content property for low-complexity H.265/HEVC encoder. IEEE Trans. Broadcast. (2016). doi:10.1109/TBC.2016.2580920
7. Pan, Z, Zhang, Y, Kwong, S: Efficient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Trans. Broadcast. 61, 166-176 (2015)
8. Xia, Z, Wang, X, Sun, X, Wang, Q: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. 27, 340-352 (2015)
9. Xia, Z, Wang, X, Sun, X, Liu, Q, Xiong, N: Steg analysis of LSB matching using differences between nonadjacent pixels. Multimed. Tools Appl. 75, 1947-1962 (2016)
10. Xia, Z, Wang, X, Zhang, L, Qin, Z, Sun, X, Ren, K: A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. (2016). doi:10.1109/TIFS.2016.2590944
11. Yuan, GL: A new method with descent property for symmetric nonlinear equations. Numer. Funct. Anal. Optim. 31, 974-987 (2010)
12. Yuan, GL, Lu, S, Wei, ZX: A new trust-region method with line search for solving symmetric nonlinear equations. Int. J. Comput. Math. 88, 2109-2123 (2011)
13. Yuan, C, Sun, X, Lv, R: Fingerprint liveness detection based on multi-scale LFPQ and PCA. China Commun. 13, 60-65 (2016)
14. Yuan, GL, Lu, XW, Wei, ZX: BFGS trust-region method for symmetric nonlinear equations. J. Comput. Appl. Math. 230, 44-58 (2009)
15. Yuan, G, Wei, Z: A trust region algorithm with conjugate gradient technique for optimization problems. Numer. Funct. Anal. Optim. 32, 212-232 (2011)
16. Yuan, GL, Wei, ZX, Lu, S: Limited memory BFGS method with backtracking for symmetric nonlinear equations. Math. Comput. Model. 54, 367-377 (2011)
17. Yuan, GL, Wei, ZX, Lu, XW: A BFGS trust-region method for nonlinear equations. Computing 92, 317-333 (2011)
18. Yuan, GL, Wei, ZX, Wang, ZX: Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minimization. Comput. Optim. Appl. 54, 45-64 (2013)
19. Yuan, GL, Wei, ZX, Wu, YL: Modified limited memory BFGS method with nonmonotone line search for unconstrained optimization. J. Korean Math. Soc. 47, 767-788 (2010)
20. Yuan, GL, Yao, SW: A BFGS algorithm for solving symmetric nonlinear equations. Optimization 62, 45-64 (2013)
21. Zhou, Z, Wang, Y, Wu, QM, Yang, C-N, Sun, X: Effective and efficient global context verification for image copy detection. IEEE Trans. Inf. Forensics Secur. (2016). doi:10.1109/TIFS.2016.2601065
22. Andrei, N: A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan. Stud. Inform. Control 17, 55-70 (2008)
23. Dai, Y: A nonmonotone conjugate gradient algorithm for unconstrained optimization. J. Syst. Sci. Complex. 15, 139-145 (2002)
24. Dai, Y, Yuan, Y: A nonlinear conjugate gradient with a strong global convergence properties. SIAM J. Optim. 10, 177-182 (2000)
25. Dai, Y, Yuan, Y: Nonlinear Conjugate Gradient Methods. Shanghai Sci. Technol, Shanghai (1998)
26. Dai, Y, Yuan, Y: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. 103, 33-47 (2001)
27. Fletcher, R: Practical Method of Optimization, Vol I: Unconstrained Optimization, 2nd edn. Wiley, New York (1997)
28. Fletcher, R, Reeves, C: Function minimization by conjugate gradients. Comput. J. 7, 149-154 (1964)
29. Grippo, L, Lucidi, S: A globally convergent version of the Polak-Ribiére gradient method. Math. Program. 78, 375-391 (1997)
30. Hager, WW, Zhang, H: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16, 170-192 (2005)
31. Hager, WW, Zhang, H: Algorithm 851: CG-DESCENT, a conjugate gradient method with guaranteed descent. ACM Trans. Math. Softw. 32, 113-137 (2006)
32. Hager, WW, Zhang, H: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. 2, 35-58 (2006)
33. Hestenes, MR, Stiefel, E: Method of conjugate gradient for solving linear equations. J. Res. Natl. Bur. Stand. 49, 409-436 (1952)
34. Polak, E, Ribiére, G: Note sur la convergence de directions conjugees. Rev. Fr. Autom. Inform. Rech. Opér. 3, 35-43 (1969)
35. Powell, M.J.D: Nonconvex minimization calculations and the conjugate gradient method. In: Numerical Analysis, Lecture Notes in Mathematics, vol. 1066, pp. 122-141. Springer, Berlin (1984)
36. Powell, M.J.D: Convergence properties of algorithm for nonlinear optimization. SIAM Rev. 28, 487-500 (1986)
37. Wei, Z, Li, G, Qi, L: New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems. Appl. Math. Comput. 179, 407-430 (2006)
38. Wei, Z, Li, G, Qi, L: Global convergence of the PRP conjugate gradient methods with inexact line search for nonconvex unconstrained optimization problems. Math. Comput. 77, 2173-2193 (2008)
39. Wei, Z, Yao, S, Liu, L: The convergence properties of some new conjugate gradient methods. Appl. Math. Comput. 183, 1341-1350 (2006)
40. Yuan, G: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. Optim. Lett. 3, 11-21 (2009)
41. Yuan, G, Duan, X, Liu, W, Wang, X, Cui, Z, Sheng, Z: Two new PRP conjugate gradient algorithms for minimization optimization models. PLoS ONE 10(10), e0140071 (2015)
42. Yuan, G, Lu, X, Wein, Z: A modified PRP conjugate gradient method. Ann. Oper. Res. 166, 73-90 (2009)
43. Yuan, G, Lu, X, Wein, Z: A conjugate gradient method with descent direction for unconstrained optimization. J. Comput. Appl. Math. 233, 519-530 (2009)
44. Yuan, G, Wei, ZX: New line search methods for unconstrained optimization. J. Korean Stat. Soc. 38, 29-39 (2009)
45. Yuan, G, Wei, ZX, Zhao, QH: A modified Polak-Ribière-Polyak conjugate gradient algorithm for large-scale optimization problems. IEE Trans. 46, 397-413 (2014)
46. Yuan, G, Zhang, MJ: A modified Hestenes-Stiefel conjugate gradient algorithm for large-scale optimization. Numer. Funct. Anal. Optim. 34, 914-937 (2013)
47. Yuan, G, Meng, ZH, Li, Y: A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. J. Optim. Theory Appl. 168, 129-152 (2016)
48. Yuan, G, Wei, ZX: The Barzilai and Borwein gradient method with nonmonotone line search for nonsmooth convex optimization problems. Math. Model. Anal. 17, 203-216 (2012)
49. Yuan, G, Wei, ZX, Li, GY: A modified Polak-Ribière-Polyak conjugate gradient algorithm for nonsmooth convex programs. J. Comput. Appl. Math. 255, 86-96 (2014)
50. Yuan, G, Zhang, MJ: A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations. J. Comput. Appl. Math. 286, 186-195 (2015)
51. Polak, E: The conjugate gradient method in extreme problems. Comput. Math. Math. Phys. 9, 94-112 (1969)
52. Yuan, G, Wei, ZX, Lu, XW: Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search. Appl. Math. Model. (2017). doi:10.1016/j.apm.2017.02.008
53. Al-Baali, A: Descent property and global convergence of the Fletcher-Reeves method with inexact line search. IMA J. Numer. Anal. 5, 121-124 (1985)
54. Gilbert, JC, Nocedal, J: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. 2, 21-42 (1992)
55. Ahmed, T, Storey, D: Efficient hybrid conjugate gradient techniques. J. Optim. Theory Appl. 64, 379-394 (1990)
56. Hu, YF, Storey, C: Global convergence result for conjugate method. J. Optim. Theory Appl. 71, 399-405 (1991)
57. Bongartz, I, Conn, AR, Gould, N, Toint, PL: CUTE: constrained and unconstrained testing environment. ACM Trans. Math. Softw. 21, 123-160 (1995)
58. Gould, N, Orban, D, Toint, PL: CUTEr and SifDec: a constrained and unconstrained testing environment, revised. ACM Trans. Math. Softw. 29, 373-394 (2003)
59. Dolan, ED, More, JJ: Benchmarking optimization software with performance profiles. Math. Program. 91, 201-213 (2002)