TAIWANESE JOURNAL OF MATHEMATICS
Vol. 27, No. 5, pp. 1021–1039, October 2023
DOI: 10.11650/tjm/230503

A Three-term Conjugate Gradient Method with a Random Parameter for Large-scale Unconstrained Optimization and its Application in Regression Model

Yueting Yang, Xue Zhang, Guoling Zhou and Mingyuan Cao*

Abstract. In this paper, a new three-term conjugate gradient algorithm is proposed to solve unconstrained optimization including regression problems. We minimize the distance between the search direction matrix and the self-scaling memoryless BFGS direction matrix in the Frobenius norm to determine the search direction, which has the same advantages as the quasi-Newton method. At the same time, random parameter is used so that the search direction satisfies sufficient descent condition. For uniformly convex functions and general nonlinear functions, we all establish the global convergence of the new method. Numerical experiments show that our method has nice numerical performance for solving large-scale unconstrained optimization. In addition, the application of the new method to the regression model proves that our method is effective.

1. Introduction

Unconstrained optimization widely exist in compressing sensing [1], portfolio selection [13] and image restoration [18,19] and many other fields. Consider the following unconstrained optimization

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

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuous differentiable and bounded from below. Starting from an initial point $x_0 \in \mathbb{R}^n$, the iterative form of conjugate gradient method is given by

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

in which $\alpha_k > 0$ is a stepsize, and d_k is a search direction defined by

$$d_0 = -g_0, \quad d_k = -g_k + \beta_{k-1} d_{k-1}, \quad k \geq 1,$$

where $g_k = \nabla f(x_k)$ is the gradient of $f(x)$ at iterate point x_k and $\beta_{k-1} \in \mathbb{R}$ is a conjugate gradient parameter.

Received January 6, 2023; Accepted May 24, 2023.
Communicated by Jein-Shan Chen.
2020 *Mathematics Subject Classification*. 90C30, 65K05.
Key words and phrases. random parameter, three-term conjugate gradient, global convergence, large-scale unconstrained optimization, regression model.
*Corresponding author.

Due to its simplicity and low memory requirement, conjugate gradient methods are the most popular class of algorithm for solving large-scale unconstrained optimization in industry and engineering field [7, 24, 25]. Recently, based on the classical conjugate gradient methods, such as DY, HS, PRP and FR, etc., many modified conjugate gradient methods are proposed [8, 20, 22, 23]. Especially, three-term conjugate gradient methods have been paid attention [2, 3, 9, 27].

Perry [21] presented the following search direction based on Dai–Liao (DL) conjugate gradient method [11]

$$d_{k+1} = -Q_{k+1} g_{k+1},$$

where Q_{k+1} is the search direction matrix, i.e.,

$$Q_{k+1} = I - \frac{s_k y_k^T}{s_k^T y_k} + t \frac{s_k s_k^T}{s_k^T y_k}, \tag{1.2}$$

$s_k = x_{k+1} - x_k = \alpha_k d_k$ and $y_k = g_{k+1} - g_k$. It is clear that Perry method and DL conjugate gradient method are equivalent. There are some different choices of parameter t [6, 26, 30].

Note that Q_{k+1} in (1.2) is nonsymmetric and does not satisfy the secant condition, Babaie-Kafaki and Ghanbari [5] presented the following new symmetric matrix

$$A_{k+1} = I - \frac{1}{2} \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + t \frac{s_k s_k^T}{s_k^T y_k}. \tag{1.3}$$

They derived $t_k^{p,q} = p \frac{\|y_k\|^2}{s_k^T y_k} - q \frac{s_k^T y_k}{\|s_k\|^2}$ ($p > 1/4$, $q \le 1/4$) by finding the eigenvalues of A_{k+1}. Their method can be regarded as a general form of methods proposed by Dai and Kou [10] and Hager and Zhang [17]. Moreover, Zhang, Liu and Liu [29] gave $t_k = \frac{\|y_k\|^2}{s_k^T y_k} - \frac{1}{4} \frac{s_k^T y_k}{\|s_k\|^2}$ by minimizing the upper bound of spectral condition number of Q_{k+1}.

In this paper, we design a new three-term conjugate gradient method with random parameter by using quasi-Newton method to derive another representation of the parameter t in (1.3). Our innovations mainly include the following:

⋄ The parameter t in A_{k+1} is determined by minimizing the distance between A_{k+1} and self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno (BFGS) matrix in the Frobenius norm.

⋄ The significant difference from previous algorithms is that random parameter is introduced to simplify the derived parameter. Our method with random parameter is more relaxed and elastic.

⋄ Our method has global convergence for uniformly convex functions and general nonlinear functions, and the method has been successfully applied to regression models.

The rest of this paper is organized as follows. In the next section, a new random parameter is given to present a three-term conjugate gradient method. In Section 3, global convergence of our method is proved under appropriate conditions. In Section 4, some numerical experiments are implemented. In Section 5, the application of the new method in regression model is presented. Conclusions are made in the last section.

2. Three-term conjugate gradient method with a random parameter

In this section, our main aim is to propose a new three-term conjugate gradient method. The search direction is derived by minimizing the Frobenius norm of difference between the search direction matrix and quasi-Newton updating, in conjunction with choices of random parameter. Inspired by Yao and Ning [27], we consider the following model

$$\min \|D_{k+1}\|_F^2, \tag{2.1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $D_{k+1} = A_{k+1} - B_{k+1}^{-1}$, A_{k+1} is determined by (1.3), B_{k+1}^{-1} is a self-scaling memoryless BFGS matrix

$$B_{k+1}^{-1} = \frac{1}{\theta_k} I - \frac{1}{\theta_k} \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + \left(1 + \frac{1}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{s_k s_k^T}{s_k^T y_k}, \tag{2.2}$$

and θ_k is a scaling parameter. From (1.3) and (2.2), we have

$$\|D_{k+1}\|_F^2 = \operatorname{tr}(D_{k+1}^T D_{k+1}) = \frac{\|s_k\|^4}{(s_k^T y_k)^2} t^2 + 2\left[\frac{\|s_k\|^2}{\theta_k s_k^T y_k} - \frac{\|s_k\|^4}{(s_k^T y_k)^2} - \frac{\|y_k\|^2 \|s_k\|^4}{\theta_k (s_k^T y_k)^3}\right] t + \xi,$$

where ξ is a constant independent of t. This is a second-degree polynomial of variable t and the coefficient of t^2 is positive. Therefore, the minimum of problem (2.1) is

$$t_k = \arg\min \left\{ \operatorname{tr}(D_{k+1}^T D_{k+1}) \right\} = 1 + \frac{m_k}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}, \tag{2.3}$$

where $m_k = \sin^2 \eta_k$, $\eta_k = \langle s_k, y_k \rangle$ is the angle between s_k and y_k. Instead of the mean value to $\cos^2 \eta_k = 1/2$ in [28], we set m_k is a random number in the interval $[\underline{c}, \overline{c}]$, where $0 < \underline{c} < \overline{c} < 1$. There are many possible ways to choose θ_k, we prefer to use

$$\theta_k = \min\left\{2\underline{c}, \frac{s_k^T y_k}{\|s_k\|^2}\right\} \quad \text{or} \quad \theta_k = \min\left\{2\underline{c}, \frac{\|y_k\|^2}{s_k^T y_k}\right\}. \tag{2.4}$$

Thus, t_k in (2.3) can be regarded as a random parameter.

Substitute (2.3) into (1.3), let $d_{k+1} = -A_{k+1} g_{k+1}$, then

$$d_{k+1} = -\left(I - \frac{1}{2} \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + t_k \frac{s_k s_k^T}{s_k^T y_k}\right) g_{k+1} \triangleq -g_{k+1} + a_k s_k + b_k y_k, \tag{2.5}$$

where

$$a_k = \frac{1}{2} \frac{y_k^T g_{k+1}}{s_k^T y_k} - \left(1 + \frac{m_k}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{s_k^T g_{k+1}}{s_k^T y_k}, \quad b_k = \frac{1}{2} \frac{s_k^T g_{k+1}}{s_k^T y_k}.$$

Based on the above analysis, a new three-term conjugate gradient algorithm with random parameter can be presented as follows.

Algorithm 2.1.

Step 0. *Given* $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $0 < \underline{c} < \bar{c} < 1$ *and* $0 < \rho < \sigma < 1$. *Let* $f_0 = f(x_0)$, $g_0 = \nabla f(x_0)$, $d_0 := -g_0$ *and* $k := 0$.

Step 1. *If* $\|g_k\| \leq \varepsilon$, *stop, else go to Step 2.*

Step 2. *Compute a steplength* α_k *satisfying strong Wolfe line search conditions*

$$f(x_k + \alpha d_k) - f(x_k) \leq \rho \alpha g_k^T d_k, \tag{2.6}$$

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k. \tag{2.7}$$

Step 3. *Set* $x_{k+1} = x_k + \alpha_k d_k$, *and compute* f_{k+1}, g_{k+1}, s_k *and* y_k.

Step 4. *Compute* t_k *by* (2.3) *and search direction* d_{k+1} *by* (2.5). *Set* $k := k + 1$ *and go to Step 1.*

The following lemma show the sufficient descent property of search direction.

Lemma 2.2. *Let the sequence* $\{d_{k+1}\}$ *be generated by Algorithm* 2.1, *then there exists a positive constant c, such that*

$$g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2. \tag{2.8}$$

Proof. From (2.5), it can be deduced that

$$\begin{aligned}
g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + \frac{y_k^T g_{k+1} g_{k+1}^T s_k s_k^T y_k}{(s_k^T y_k)^2} - \left(1 + \frac{m_k}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{(s_k^T g_{k+1})^2}{s_k^T y_k} \\
&\leq -\|g_{k+1}\|^2 + \frac{1}{2} \frac{(g_{k+1}^T s_k)^2 \|y_k\|^2 + (s_k^T y_k)^2 \|g_{k+1}\|^2}{(s_k^T y_k)^2} \\
&\quad - \left(1 + \frac{m_k}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{(s_k^T g_{k+1})^2}{s_k^T y_k} \\
&= -\frac{1}{2}\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{s_k^T y_k} \left[1 + \frac{\|y_k\|^2}{s_k^T y_k} \left(\frac{m_k}{\theta_k} - \frac{1}{2}\right)\right] \\
&\leq -\frac{1}{2}\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{s_k^T y_k} \left[1 + \frac{\|y_k\|^2}{s_k^T y_k} \left(\frac{2\underline{c} - \theta_k}{2\theta_k}\right)\right] \\
&\leq -\frac{1}{2}\|g_{k+1}\|^2.
\end{aligned}$$

The second of above inequality is from the fact $u^T v \leq \frac{1}{2}(\|u\|^2 + \|v\|^2)$ in which $u = g_{k+1}^T s_k y_k$ and $v = s_k^T y_k g_{k+1}$. It is well known that $s_k^T y_k > 0$ can be ensured by Wolfe line search. Combining (2.4), let $c = 1/2$, the proof is completed. □

3. Convergence analysis

In this section, to prove the global convergence of Algorithm 2.1, we give the following assumptions.

Assumption 3.1. *The level set $\Omega = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$ is bounded, namely, there exists a positive constant δ such that $\|x\| \le \delta$, $\forall\, x \in \Omega$.*

Assumption 3.2. *The gradient of function f is Lipschitz continuous in some neighborhood $\mathbb{N}$ of Ω, namely, there exists $L > 0$ satisfying*

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall\, x, y \in \mathbb{N}. \tag{3.1}$$

Based on the above assumptions, we can easily see that $g(x)$ is bounded, namely, there exists a positive constant M such that

$$\|g(x)\| \le M, \quad \forall\, x \in \Omega. \tag{3.2}$$

Lemma 3.3. *Let the sequence $\{d_k\}$ be generated by Algorithm 2.1. If Assumption 3.2 holds, then*

$$\alpha_k \ge \frac{(1-\sigma)|g_k^T d_k|}{L\|d_k\|^2}.$$

Proof. The proof of Lemma 3.3 is similar to that of Proposition 4.1 in [2], so we omit it here. □

Lemma 3.4. *Let the sequence $\{d_k\}$ be generated by Algorithm 2.1. If Assumption 3.2 holds, we have*

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

Proof. From the first inequality (2.6) of strong Wolfe conditions, Assumption 3.2 and Lemma 3.3, we have

$$f_k - f_{k+1} \ge -\rho \alpha_k g_k^T d_k \ge -\rho \frac{(1-\sigma)(g_k^T d_k)^2}{L\|d_k\|^2}.$$

Since $f(x)$ is bounded from below, (3.3) is obtained. □

Theorem 3.5. *Suppose that Assumptions 3.1 and 3.2 hold. The sequence $\{x_k\}$ is generated by Algorithm 2.1. If f is a uniformly convex function on Ω, namely, there exists a positive constant μ such that*

$$(\nabla f(x) - \nabla f(y))^T (x - y) \ge \mu\|x - y\|^2, \quad \forall\, x, y \in \mathbb{N}, \tag{3.4}$$

then we have

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

Proof. From the Lipschitz condition (3.1), we have

$$\|y_k\| = \|g_{k+1} - g_k\| \le L\|s_k\|. \tag{3.5}$$

It follows from (3.4) that

$$y_k^T s_k \ge \mu\|s_k\|^2. \tag{3.6}$$

Using Cauchy inequality and (3.6), we obtain $\mu\|s_k\|^2 \le y_k^T s_k \le \|y_k\|\|s_k\|$, namely,

$$\mu\|s_k\| \le \|y_k\|. \tag{3.7}$$

Then, from (3.5), (3.6) and (3.7), we have

$$\begin{aligned} \mu = \frac{\mu\|s_k\|^2}{\|s_k\|^2} &\le \frac{s_k^T y_k}{\|s_k\|^2} \le \frac{\|y_k\|\|s_k\|}{\|s_k\|^2} \le L, \\ \mu \le \frac{\|y_k\|^2}{\|y_k\|\|s_k\|} &\le \frac{\|y_k\|^2}{s_k^T y_k} \le \frac{\|y_k\|^2}{\mu\|s_k\|^2} \le \frac{L^2}{\mu}. \end{aligned} \tag{3.8}$$

Let $\theta_{\min} = \min\{2\underline{c}, \mu\}$, we get $\theta_k \ge \theta_{\min}$. From (3.8), we obtain

$$t_k = 1 + \frac{m_k}{\theta_k}\frac{\|y_k\|^2}{s_k^T y_k} \le 1 + \frac{\bar{c}}{\theta_{\min}}\frac{L^2}{\mu}. \tag{3.9}$$

Therefore, by the definition of d_k, triangle inequality, Cauchy inequality, (3.5), (3.6) and (3.9), we have

$$\begin{aligned} \|d_{k+1}\| &= \|-g_{k+1} + a_k s_k + b_k y_k\| \\ &\le \|g_{k+1}\| + \frac{1}{2}\left|\frac{s_k^T g_{k+1}}{s_k^T y_k}\right|\|y_k\| + \frac{1}{2}\left|\frac{y_k^T g_{k+1}}{s_k^T y_k}\right|\|s_k\| + |t_k|\left|\frac{s_k^T g_{k+1}}{s_k^T y_k}\right|\|s_k\| \\ &\le \|g_{k+1}\| + \frac{1}{2}\frac{\|g_{k+1}\|\|y_k\|}{\mu\|s_k\|} + \frac{1}{2}\frac{\|g_{k+1}\|\|y_k\|}{\mu\|s_k\|} + \left(1 + \frac{\bar{c}}{\theta_{\min}}\frac{L^2}{\mu}\right)\frac{\|g_{k+1}\|}{\mu} \\ &\le \left(1 + \frac{L+1}{\mu} + \frac{\bar{c}}{\theta_{\min}}\frac{L^2}{\mu^2}\right)\|g_{k+1}\| \triangleq M_1\|g_{k+1}\|. \end{aligned} \tag{3.10}$$

From Lemma 2.2 and (3.10), we know

$$\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} \ge \frac{c^2\|g_{k+1}\|^2}{M_1^2}.$$

Combine with Lemma 3.4, then

$$\sum_{k=0}^{\infty}\|g_k\|^2 < \infty.$$

The proof is completed. □

For general nonlinear functions, we can establish a weaker convergence result

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.11}$$

Lemma 3.6. *Suppose that Assumptions* 3.1 *and* 3.2 *hold. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*, then we have* $d_k \neq 0$ *and*

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty, \tag{3.12}$$

whenever $\inf\{\|g_k\| : k \geq 0\} > 0$ *in which* $u_k = d_k/\|d_k\|$.

Proof. Define $\gamma = \inf\{\|g_k\| : k \geq 0\}$, then $\|g_k\| \geq \gamma > 0$. From the sufficient descent condition (2.8), we know $d_k \neq 0$ for each k, so u_k is well defined. To prove global convergence, we define $a_k^+ = \max\{a_k', 0\}$, where $a_k' = \frac{1}{2}\frac{y_k^T g_{k+1}}{d_k^T y_k} - \left(1 + \frac{m_k}{\theta_k}\frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{s_k^T g_{k+1}}{d_k^T y_k}$. By (2.5), we have

$$\frac{d_{k+1}}{\|d_{k+1}\|} = \frac{-g_{k+1}}{\|d_{k+1}\|} + a_k^+ \frac{d_k}{\|d_{k+1}\|} + b_k \frac{y_k}{\|d_{k+1}\|} = \frac{-g_{k+1} + b_k y_k}{\|d_{k+1}\|} + a_k^+ \frac{\|d_k\|}{\|d_{k+1}\|}\frac{d_k}{\|d_k\|},$$

namely,

$$u_{k+1} = \omega_k + \delta_k u_k,$$

where

$$\omega_k = \frac{-g_{k+1} + b_k y_k}{\|d_{k+1}\|}, \quad \delta_k = a_k^+ \frac{\|d_k\|}{\|d_{k+1}\|} \geq 0.$$

Using the identity $\|u_{k+1}\| = \|u_k\| = 1$,

$$\|\omega_k\| = \|u_{k+1} - \delta_k u_k\| = \|\delta_k u_{k+1} - u_k\|.$$

Since $\delta_k \geq 0$, it follows that

$$\|u_{k+1} - u_k\| \leq \|(1+\delta_k)u_{k+1} - (1+\delta_k)u_k\| \leq \|u_{k+1} - \delta_k u_k\| + \|\delta_k u_{k+1} - u_k\| = 2\|\omega_k\|.$$

From (2.7), we have

$$\left|\frac{s_k^T g_{k+1}}{s_k^T y_k}\right| = \left|\frac{d_k^T g_{k+1}}{d_k^T y_k}\right| \leq \frac{\sigma}{1-\sigma}, \quad \|y_k\| \leq \|g_{k+1}\| + \frac{\|g_k\|}{\|g_{k+1}\|}\|g_{k+1}\| \leq 1 + \frac{M}{\gamma}\|g_{k+1}\|. \tag{3.13}$$

By the definition of ω_k, b_k and (3.13), we get

$$\begin{aligned}\|\omega_k\| &= \frac{\|-g_{k+1} + b_k y_k\|}{\|d_{k+1}\|} \\ &\leq \frac{\|g_{k+1}\| + \frac{1}{2}\left|\frac{s_k^T g_{k+1}}{s_k^T y_k}\right| \cdot \|y_k\|}{\|d_{k+1}\|} \leq \left[1 + \frac{\sigma}{2(1-\sigma)}\left(1 + \frac{M}{\gamma}\right)\right]\frac{\|g_{k+1}\|}{\|d_{k+1}\|}.\end{aligned}$$

If $\|g_{k+1}\| > \gamma$, from Lemmas 2.2 and 3.4, we have

$$\sum_{k=0}^{\infty} \frac{c^2\gamma^2\|g_{k+1}\|^2}{\|d_{k+1}\|^2} \leq \sum_{k=0}^{\infty} \frac{c^2\|g_{k+1}\|^4}{\|d_{k+1}\|^2} \leq \sum_{k=0}^{\infty} \frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} < +\infty,$$

therefore, (3.12) holds. □

Definition 3.7. Consider a method of the form (1.1) and (2.5), and suppose

$$0 < \gamma \leq \|g_k\| \leq \overline{\gamma}, \quad k \geq 0. \tag{3.14}$$

We call that a method has Property (*) if there exist constants $b > 1$ and $\lambda > 0$ such that $|a'_k| < b$ and $\|s_k\| \leq \lambda$, then $|a'_k| \leq \frac{1}{2b}$.

Lemma 3.8. *Suppose that Assumptions* 3.1 *and* 3.2 *hold. Let the sequence* $\{d_k\}$ *be generated by Algorithm* 2.1, *then Algorithm* 2.1 *has Property (*).*

Proof. By (2.7) and (2.8), we obtain

$$d_k^T y_k \geq (\sigma - 1) g_k^T d_k \geq c(1-\sigma)\|g_k\|^2. \tag{3.15}$$

Using (3.2), (3.14), Assumption 3.1 and (3.15), we obtain

$$\begin{aligned}
|a'_k| &= \left| \frac{1}{2}\frac{y_k^T g_{k+1}}{d_k^T y_k} - \left(1 + \frac{m_k}{\theta_k}\frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{s_k^T g_{k+1}}{d_k^T y_k} \right| \\
&\leq \frac{1}{2}\frac{\|y_k\|\|g_{k+1}\|}{c(1-\sigma)\|g_k\|^2} + \left(1 + \frac{m_k}{\theta_k}\frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{\|s_k\|\|g_{k+1}\|}{c(1-\sigma)\|g_k\|^2} \\
&\leq \frac{1}{2}\frac{\|g_{k+1}-g_k\|\|g_{k+1}\|}{c(1-\sigma)\|g_k\|^2} + \left(1 + \frac{1}{\theta_{\min}}\frac{\|g_{k+1}-g_k\|^2}{c(1-\sigma)\|g_k\|^2}\right)\frac{\|s_k\|\|g_{k+1}\|}{c(1-\sigma)\|g_k\|^2} \\
&\leq \frac{\overline{\gamma}^2}{c(1-\sigma)\gamma^2} + \left(1 + \frac{2}{\theta_{\min}}\frac{2\overline{\gamma}^2}{c(1-\sigma)\gamma^2}\right)\frac{2\delta\overline{\gamma}}{c(1-\sigma)\gamma^2} := b.
\end{aligned} \tag{3.16}$$

Define

$$\lambda := \frac{c^2(1-\sigma)^2\gamma^4}{2\overline{\gamma}^2\left[\overline{\gamma} + \left(1 + \frac{4\delta}{\theta_{\min}}\frac{2\overline{\gamma}^2}{c(1-\sigma)\gamma^2}\right)\right]\left[\frac{L}{2} + \left(1 + \frac{2}{\theta_{\min}}\frac{2\overline{\gamma}^2}{c(1-\sigma)\gamma^2}\right)\right]}, \tag{3.17}$$

if $\|s_k\| \leq \lambda$, from (3.16) and (3.17), we obtain

$$\begin{aligned}
|a'_k| &\leq \frac{1}{2}\frac{L\|s_k\|\|g_{k+1}\|}{c(1-\sigma)\gamma^2} + \left(1 + \frac{1}{\theta_{\min}}\frac{L^2\|s_k\|^2}{c(1-\sigma)\gamma^2}\right)\frac{\|s_k\|\|g_{k+1}\|}{c(1-\sigma)\gamma^2} \\
&\leq \left[\frac{1}{2}\frac{L\overline{\gamma}}{c(1-\sigma)\gamma^2} + \left(1 + \frac{2}{\theta_{\min}}\frac{2\overline{\gamma}^2}{c(1-\sigma)\gamma^2}\right)\frac{\overline{\gamma}}{c(1-\sigma)\gamma^2}\right]\|s_k\| \\
&\leq \left[\frac{1}{2}\frac{L\overline{\gamma}}{c(1-\sigma)\gamma^2} + \left(1 + \frac{2}{\theta_{\min}}\frac{2\overline{\gamma}^2}{c(1-\sigma)\gamma^2}\right)\frac{\overline{\gamma}}{c(1-\sigma)\gamma^2}\right]\lambda = \frac{1}{2b}.
\end{aligned}$$

□

We will show that if the gradient sequence is bounded away from zero, then a fraction of the steps cannot be too small in the next lemma. Let $\mathbb{N}$ be the set of positive integers. Let $\mathcal{K}^\lambda := \{i \in \mathbb{N} : i \geq 1, \|s_i\| > \lambda\}$ for $\lambda > 0$, namely, the set of integers corresponding to steps greater than λ. Now, we need to discuss groups of $\triangle$ consecutive iterates, we let $\mathcal{K}^\lambda_{k,\Delta} := \{i \in \mathbb{N} : k \leq i \leq k + \Delta - 1, \|s_i\| > \lambda\}$. Let $|\mathcal{K}^\lambda_{k,\Delta}|$ denote the number of elements of $\mathcal{K}^\lambda_{k,\Delta}$.

Lemma 3.9. *Suppose that Assumptions* 3.1 *and* 3.2 *hold. Let the sequences* $\{x_k\}$ *and* $\{d_k\}$ *be generated by Algorithm* 2.1. *When* (3.14) *holds, there exists* $\lambda > 0$ *such that*

$$\left|\mathcal{K}^\lambda_{k,\Delta}\right| > \frac{\Delta}{2} \quad \textit{for } \Delta \in \mathbb{N},$$

where $k \geq k_0$ *with* k_0 *being the any index.*

Proof. We proceed by contradiction. Suppose that

$$\begin{aligned}&\text{for any } \lambda > 0, \text{ there exist } \Delta \in \mathbb{N} \text{ and } k_0 \text{ such that,}\\ &\text{for any } k \geq k_0, \text{ we have } \left|\mathcal{K}^\lambda_{k,\Delta}\right| \leq \Delta/2.\end{aligned} \tag{3.18}$$

By (3.13), we have

$$\|b_k y_k\| = \frac{1}{2}\left|\frac{s_k^T g_{k+1}}{s_k^T y_k}\right| \|y_k\| \leq \frac{\sigma}{2(1-\sigma)\left(1+\frac{\overline{\gamma}}{\gamma}\right)\|g_{k+1}\| \triangleq M_2\|g_{k+1}\|}.$$

According to the definition of (2.5), we have

$$\begin{aligned}\|d_{k+1}\|^2 &\leq \left(a'_k\|d_k\| + \|-g_{k+1} + b_k y_k\|\right)^2 \leq 2a'^2_k\|d_k\|^2 + 2\|-g_{k+1} + b_k y_k\|^2\\ &\leq 2a'^2_k\|d_k\|^2 + 2\left(2\|g_{k+1}\|^2 + 2\|b_k y_k\|^2\right) \leq 2a'^2_k\|d_k\|^2 + 4(1+M_2^2)\|g_{k+1}\|^2,\end{aligned}$$

the above inequalities are established based on $2ab \leq a^2 + b^2$ for any scalars a and b, so $(a+b)^2 \leq 2a^2 + 2b^2$. By induction, we have

$$\|d_l\|^2 \leq c_1\left(1 + 2a'^2_{l-1} + 2a'^2_{l-1}2a'^2_{l-2} + \cdots + 2a'^2_{l-1}2a'^2_{l-2}\cdots 2a'^2_{k_0}\right) \tag{3.19}$$

for any given index $l \geq k_0 + 1$, where c_1 depends on $\|d_{k_0-1}\|$, not depends on l. Next, we consider $2a'^2_{l-1}2a'^2_{l-2}\cdots 2a'^2_k$, where $k_0 \leq k \leq l-1$. We divide $2(l-k)$ factors of (3.19) into groups of each 2Δ elements, namely, if $\Lambda := (l-k)/\Delta$, then (3.19) can be divided into Λ or $\Lambda + 1$ groups

$$(2a'^2_{l_1}\cdots 2a'^2_{k_1}),\ \ldots,\ (2a'^2_{l_\Lambda}\cdots 2a'^2_{k_\Lambda}), \tag{3.20}$$

and a possible group

$$(2a'^2_{l_\Lambda+1}\cdots 2a'^2_{k}), \tag{3.21}$$

where $l_i = l - 1 - (i-1)\Delta$ for $i = 1, 2, \ldots, \Lambda + 1$, and $k_i = l_{i+1} + 1$ for $i = 1, 2, \ldots, \Lambda$. It is clear that $k_i \geq k_0$ for $i = 1, 2, \ldots, \Lambda$, from (3.18), we get $p_i := \left|\mathcal{K}^{\lambda}_{k_i,\Delta}\right| \leq \Delta/2$. Thus, there are p_i indices j such that $\|s_j\| > \lambda$ and $(\Delta - p_i)$ indices j such that $\|s_j\| \leq \lambda$ on $[k_i, k_i + \Delta - 1]$.

From (3.16), we have $b > \frac{\overline{\gamma}^2}{c(1-\sigma)\gamma^2} > 1$, i.e., $2b^2 > 1$. In conjunction with $2p_i - \Delta \leq 0$, we have $2a'^2_{l_i} \cdots 2a'^2_{k_i} \leq 2^{\Delta} b^{2p_i} \left(\frac{1}{2b}\right)^{2(\Delta - p_i)} = (2b^2)^{2p_i - \Delta} \leq 1$. So every item in (3.20) is less than or equal to 1, and so is their product. In (3.21), we have $2a'^2_{l_{\Lambda+1}} \cdots 2a'^2_{k} \leq (2b^2)^{\Delta}$. Then, we get

$$\|d_l\|^2 \leq c_2(l - k_0 + 2),$$

where $c_2 > 0$ and independent of l. Furthermore, $\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty$. But from sufficient condition (2.8), Zoutendijk condition (3.3) and (3.14), we have

$$c^2\gamma^4 \sum_{k\geq 0} \frac{1}{\|d_k\|^2} \leq c^2 \sum_{k \geq 0} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.$$

This leads to a contradiction. The proof is completed. □

Theorem 3.10. *Suppose that Assumptions* 3.1 *and* 3.2 *hold. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*, then* (3.11) *holds.*

Proof. Suppose on the contrary that we can get a contradiction similarly to Theorem 4.3 in [15]. □

4. Numerical results

In this section, we show the numerical performance of Algorithm 2.1. All codes are written on Matlab R2015b and run on PC with 1.80 GHz CPU processor, 8.00 GB RAM memory. Two classes of test problems were selected here which are listed in Table 4.1. One class was drawn from the CUTEr library [16], and the other class came from Andrei [4]. Table 4.1 lists 28 test functions for 80 problems with dimensions from 2 to 50000.

Table 4.1: List of the test functions and dimensions.

	Functions		Functions		Functions		Functions
1	Freudenstein and Roth	2	Chebyquad	3	Powell badly scaled	4	Beale
5	Helical valley	6	Broyden banded	7	Wood	8	Biggs EXP6
9	Extended Rosenbrock	10	Trigonometric	11	Extended Powell singular	12	Boundary value
13	Penalty function I	14	Integral equation	15	Penalty function II	16	Broyden tridiagonal
17	Gaussian-1	18	Gaussian-2	19	Box-1	20	Box-2
21	Separable cubic	22	Variable dimension	23	Nearly separable	24	Watson
25	Yang tridiagonal	26	Brown and Dennis-1	27	Brown and Dennis-2	28	Allgower

Table 4.2: Partial numerical results of m_k with random parameter and 1/2.

$P.$	RTT1(r) (k/nf/CPU)	RTT1($\frac{1}{2}$) (k/nf/CPU)	RTT2(r) (k/nf/CPU)	RTT2($\frac{1}{2}$) (k/nf/CPU)
2	12/28/0.0469	55/85/0.0313	12/28/0.0469	64/95/0.0313
4	15/26/0.0156	15/26/0.0156	15/25/0.0313	15/25/0.0313
5	54/80/0.0469	72/102/0.0469	66/90/0.0781	72/102/0.0469
6	117/156/0.0313	145/197/0.0625	87/122/0.0156	145/197/0.0781
7	92/127/0.0156	284/363/0.0625	104/136/0.0313	131/177/0.1094
8	94/129/0.0313	102/145/0.0781	115/166/0.0313	102/145/0.0313
12	77/104/0.1250	85/116/0.0469	81/110/0.0625	133/185/0.0781
13	57/72/0.5938	121/160/0.7188	101/134/1.1094	117/153/0.6875
14	88/123/1.1750	104/144/1.3438	97/136/1.2188	131/180/1.6719
15	90/125/7.1875	157/214/8.4219	120/166/11.8906	122/167/6.1250
22	16/22/0.0010	19/26/0.0156	15/21/0.0156	24/30/0.0156
23	11/21/0.0156	13/25/0.0275	12/22/0.0010	16/24/0.0156
24	7/24/0.0012	8/24/0.0313	6/17/0.0010	8/24/0.0565
26	16/53/0.0013	16/53/0.0156	16/54/0.0015	16/53/0.0156
41	14/21/1.0313	50/66/3.7344	32/46/2.5469	50/66/3.6406
53	46/57/0.0075	52/62/0.0156	32/42/0.0156	53/66/0.0156
54	41/51/0.1250	41/51/0.1250	45/59/0.1250	63/75/0.1250
55	30/38/0.1788	42/52/0.2031	37/46/0.2813	37/46/0.2813
56	22/29/0.4219	31/40/0.5469	34/44/0.5938	52/61/0.9219
61	41/55/0.0313	41/55/0.0313	41/55/0.0156	41/55/0.0156
62	42/60/0.1406	42/60/0.1406	35/47/0.1250	42/60/0.1250
63	39/52/0.8125	46/62/0.8125	40/55/0.3281	40/55/0.3281
64	46/62/0.7813	46/62/0.7813	50/66/0.9063	46/62/0.8438
65	10/15/0.1094	12/15/0.1563	9/13/0.0938	10/14/0.1250
66	9/14/3.6250	15/18/6.0313	9/14/3.7031	11/15/4.3594
67	14/18/18.3906	16/20/22.4375	11/16/11.7500	11/15/13.9688
68	11/16/279.3438	14/18/401.2656	13/18/326.5781	13/18/373.9375
69	44/81/0.5625	44/81/0.5625	117/184/1.8438	44/81/0.5625

We compare Algorithm 2.1 against TTCG (STT) [9], NTPA [27], CG_descent (DDL) [17] and TCG [5], which have been acclaimed to be powerful for solving unconstrained optimization problems. When $\theta_k = \min\left\{2\underline{c}, \frac{s_k^T y_k}{\|s_k\|^2}\right\}$ and $\theta_k = \min\left\{2\underline{c}, \frac{\|y_k\|^2}{s_k^T y_k}\right\}$ are chosen, Algorithm 2.1 are denoted by "RTT1" and "RTT2", respectively.

All test methods are terminated when satisfies condition $\|g_k\| \le \varepsilon$ or the number of iterations exceeds 1000. We set parameters as $\varepsilon = 10^{-6}$; $\rho = 0.01$, $\sigma = 0.8$ in the strong Wlofe conditions (2.6) and (2.7); $p = 0.8$, $q = -0.1$ in CG_descent method and $\eta = 0.1$ in TCG method.

Table 4.3: Partial numerical results of several methods.

$P.$	RTT1 (k/nf/CPU)	STT (k/nf/CPU)	NTPA (k/nf/CPU)	DDL (k/nf/CPU)	TCG (k/nf/CPU)
11	12/28/0.0469	25/54/0.0156	9/25/0.0625	14/32/0.0625	30/60/0.0156
14	15/26/0.0156	15/31/0.0250	48/61/0.0781	12/21/0.0625	22/37/0.0156
15	54/80/0.0469	42/98/0.0156	$-/-/-$	57/88/0.0156	38/63/0.0469
18	117/156/0.0313	95/218/0.0156	170/189/0.0313	43/63/0.0469	25/47/0.0156
21	16/22/0.0010	11/22/0.0250	32/42/0.0781	7/11/0.0156	16/23/0.0250
28	11/21/0.0156	7/18/0.0781	31/43/0.0313	21/28/0.0156	16/25/0.0156
29	7/24/0.0010	4/11/0.0313	11/28/0.0378	8/26/0.0250	5/19/0.0010
32	16/53/0.0010	14/57/0.0250	16/53/0.0378	16/53/0.0313	16/55/0.0156
33	14/21/1.0313	23/47/2.2500	26/33/0.7188	57/84/1.7656	29/39/2.2969
43	46/57/0.0313	19/40/0.0781	63/79/0.0313	40/50/0.0625	52/64/0.0156
44	41/51/0.1250	24/48/0.0625	72/90/0.2344	33/41/0.1406	51/63/0.2500
47	30/38/0.2188	28/55/0.1875	78/91/0.3438	49/61/0.3594	53/70/0.3906
48	22/29/0.4219	29/56/0.5625	42/57/0.4531	62/76/0.7188	39/50/0.8906
51	41/55/0.0313	28/59/0.0781	38/56/0.0313	51/68/0.0313	65/86/0.0156
52	42/60/0.1406	27/55/0.0625	107/126/0.2656	38/50/0.1406	52/69/0.1250
56	39/52/1.4078	17/44/0.0625	73/89/0.3594	48/64/0.3594	45/61/0.2813
57	46/62/0.7813	88/177/2.0781	87/108/0.9688	38/60/0.3906	45/67/0.8906
63	10/15/0.1094	11/15/0.2188	12/14/0.0625	13/15/0.0781	10/14/0.1094
64	9/14/3.6250	11/15/2.0313	13/15/2.5625	18/20/3.2656	11/18/4.2031
68	14/18/18.3906	11/15/5.8438	13/15/7.7813	20/22/11.4219	16/20/20.5469
72	11/16/279.3438	13/18/177.3750	15/17/491.2188	22/24/339.5000	12/17/301.5469

Some numerical results of m_k with random parameter and 1/2 for problems (in Table 4.1) are listed in Table 4.2. And some experimental results compared with several methods are listed in Table 4.3.

As can be seen from Table 4.2, RTT1 and RTT2 with m_k taking random parameter performs better than that with m_k taking 1/2 with respect to k, nf and CPU. Therefore, RTT1 and RTT2 with random parameters are more competitive. Moreover, the performance of RTT1 is superior to that of RTT2.

From Table 4.3, if program runs failure, or the number of iterations reaches more than 500, regarded as failed (−). We can see that RTT1 algorithm is effective than other methods in most cases.

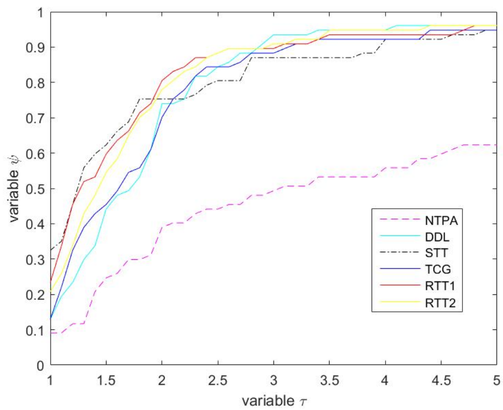

Figure 4.1: The number of iterations (k).

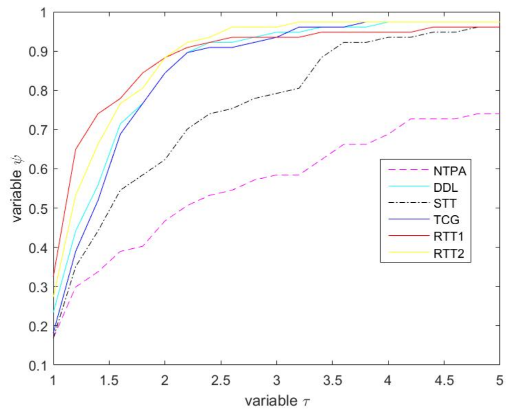

Figure 4.2: The number of function evaluations (nf).

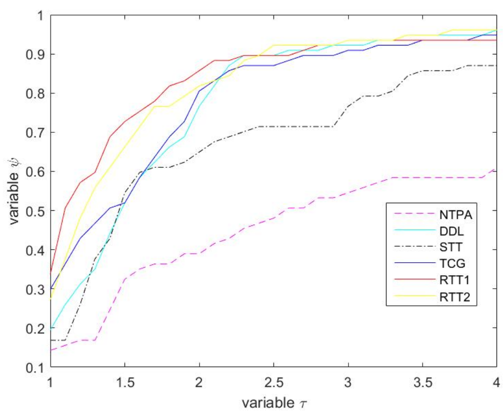

Figure 4.3: The number of gradient evaluations (ng).

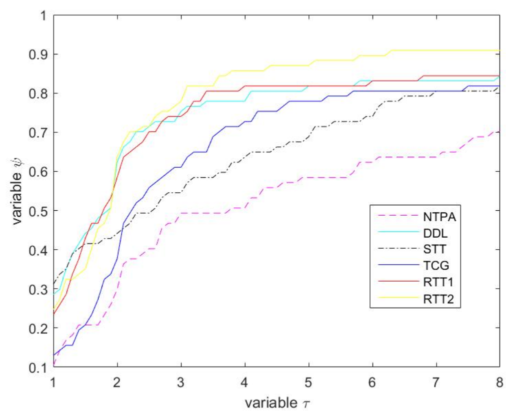

Figure 4.4: CPU time.

And we present the performance profile including (k, nf, ng and CPU time) introduced by Dolan and Moré [14] to clearly show the difference in numerical effects among six algorithms. Generally, the method whose performance profile plot is on the top right will represent the best method. Let Y and W be the set of methods and test problems, n_y, n_w be the number of methods and test problems, respectively. The performance profile $\psi\colon \mathbb{R} \to [0,1]$ is for each $y \in Y$ and $w \in W$ defined that $a_{w,y} > 0$ is k or nf or ng or CPU required to solve problems w by method y. Furthermore, the performance profile is obtained by

$$r_{w,y} = \frac{a_{w,y}}{\min\{a_{w,y} : y \in Y\}}, \quad \psi_y(\tau) = \frac{1}{n_w} \operatorname{size}\{w \in W : r_{w,y} \leq \tau\},$$

where $\tau > 0$, size$\{\cdot\}$ is the number of elements in a set.

Figures 4.1–4.4 plot the performance profiles for the number of iterations, the number of function evaluations, the number of gradient evaluations, and the CPU time, respectively. They show that RTT1 and RTT2 are superior to other algorithms when $\tau < 2.5$ and when $\tau > 2.5$. RTT1 and RTT2 are comparable to the best-performing DDL and TCG algorithms.

5. The application of Algorithm 2.1 in regression model

In this section, it is considered that using conjugate gradient method to solve regression problems can improve the solving efficiency and accuracy, process large-scale data sets, and is also widely used in other fields. We apply Algorithm 2.1 to a practical problem of regression analysis in [12] and compare our algorithm with [12] to verify the applicability. Regression analysis is one of the most commonly tools used data modeling and analysis in economics, finance and other fields. For example, it can be used to build predictive models between variables to predict future outcomes; it can help determine which factors are most important to the outcome; it can also be used to identify outliers, values that adversely affect results. So the application of Algorithm 2.1 in regression analysis has important practical significance. In Table 5.1, a summary of the number of female deaths in Irbid (Jordan) from 2009 to 2018 is given. The data set is retrieved from the Department of Statistics in Jordan (2018).

Based on the research of Dawahdeh et al. [12], the optimization problem for finding quadratic regression parameters is defined as follows:

$$\min f(a) = \sum_{i=1}^{n} \left[y_i - a(1, x_i, x_i^2)^T\right]^2, \quad a = (a_0, a_1, a_2) \in \mathbb{R}^3. \tag{5.1}$$

According to data 1 to 9 in Table 5.1, (5.1) is transformed into the following problem

$$\begin{aligned} \min f(x_1, x_2, x_3) = {}& 9x_1^2 + 90x_1x_2 + 570x_1x_3 - 33330x_1 + 285x_2^2 \\ & + 4050x_2x_3 - 176342x_2 + 15333x_3^2 - 1155094x_3 + 31282025. \end{aligned} \tag{5.2}$$

Then Algorithm 2.1 is used to calculate the optimization problem (5.2), choose different initial points, the results are shown in Table 5.2.

Table 5.1: Number of female deaths recorded in Irbid city (Jordan) from 2009 to 2018.

Number of Data (x)	Year	Number of Female Deaths (y)
1	2009	1563
2	2010	1689
3	2011	1647
4	2012	1679
5	2013	1757
6	2014	1973
7	2015	2071
8	2016	2121
9	2017	2165
10	2018	2117

Table 5.2: Test results for optimization of quadratic model for Algorithm 2.1.

Initial Point	Solution Point (a_0, a_1, a_2)
(1,1,1)	(1528.889793318067, 36.424065643404, 4.438905767442)
(9,9,9)	(1528.733929199936, 36.637097404077, 4.413085031462)
(13,13,13)	(1529.214320770277, 36.385744696179, 4.437481293816)
(1000,1000,1000)	(1529.571675208395, 36.170828337683, 4.459279814400)

By solving the average values of a_0, a_1 and a_2, we obtained the quadratic function of regression analysis as

$$\widehat{y} = 1529.102429624169 + 36.404434020335749x + 4.437187976780001x^2. \tag{5.3}$$

Now, we use the relative error $\left|\frac{y-\widehat{y}}{y}\right|$ to measure the fitting degree of (5.3) with observed data, and compared with the method proposed in [12]. The smaller the relative error value is, the better the accuracy is, or the better the fitting with the observed data set is. The results are shown in Table 5.3.

Table 5.3: Relative error for quadratic model.

Year (x)	Female Deaths (y)	$\widehat{y}$	Relative Error
1	1563	1569.944051621285	0.004442771350790
2	1689	1619.660049571961	0.041053848684452
3	1647	1678.250423476196	0.018974149044442
4	1679	1745.715173333992	0.039735064522926
5	1757	1822.054299145348	0.037025782097523
6	1973	1907.267800910263	0.033315863704884
7	2071	2001.355678628739	0.033628354114563
8	2121	2104.317932300775	0.007865189862907
9	2165	2216.154561926371	0.023627973176153

According to Table 5.3, the sum of relative errors is

$$\begin{aligned}
&0.004442771350790 + 0.041053848684452 + 0.018974149044442\\
&\ + 0.039735064522926 + 0.037025782097523 + 0.033315863704884\\
&\ + 0.033628354114563 + 0.007865189862907 + 0.023627973176153\\
= &\ 0.239668996558640.
\end{aligned}$$

And the average value of relative errors is 0.026629888506516. The sum and average relative errors of the method proposed [12] are 0.23968163237293 and 0.02663129248588, respectively. By comparison, it can be found that the sum of relative errors and average relative errors obtained by using Algorithm 2.1 to solve regression model (5.2) are lower than the algorithm in reference [12]. Therefore, we can think that the quadratic model obtained by using Algorithm 2.1 has a good fitting degree and improves the accuracy of the model.

6. Conclusion

In this paper, a class of three-term conjugate gradient methods with random parameter are proposed. The Frobenius norm is used to minimize distance between the symmetric matrix A_{k+1} and memoryless BFGS matrix, and calculation format of parameter in A_{k+1} is obtained. A random technique is to simplify parameter and a new search direction is derived which has sufficient descent property. Global convergence of new algorithm is proved under appropriate assumptions. Furthermore, some classical test problems are selected for numerical experiments and compared with the other two three-term conjugate gradient methods to verify the effectiveness of proposed algorithm. Finally, the new algorithm is applied to a practical problem in regression analysis, and shown that our algorithm is competitive.

Acknowledgments

This work is supported by the project of natural science foundation joint fund of Jilin Province (YDZJ202101ZYTS167, YDZJ202101ZYTS156 and YDZJ202201ZYTS303); the project of education department of Jilin Province (JJKH20210030KJ, JJKH20230054KJ); the Graduate Innovation Project of Beihua University (2019002, 2021002); the youth science and technology innovation team cultivation program of Beihua University. The authors are grateful to the editor and the anonymous reviewers for their valuable comments and suggestions, which have substantially improved this paper.

References

[1] A. B. Abubakar, K. Muangchoo, A. Muhammad and A. H. Ibrahim, *A spectral gradient projection method for sparse signal reconstruction in compressive sensing*, Mod. Appl. Sci. **14** (2020), no. 5, 86–93.

[2] N. Andrei, *A simple three-term conjugate gradient algorithm for unconstrained optimization*, J. Comput. Appl. Math. **241** (2013), 19–29.

[3] ______, *A Dai–Liao conjugate gradient algorithm with clustering of eigenvalues*, Numer. Algorithms **77** (2018), no. 4, 1273–1282.

[4] ______, *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*, Springer Optimization and its Applications **158**, Springer, Cham, 2020.

[5] S. Babaie-Kafaki and R. Ghanbari, *A descent family of Dai–Liao conjugate gradient methods*, Optim. Methods Softw. **29** (2014), no. 3, 583–591.

[6] S. Babaie-Kafaki, R. Ghanbari and N. Mahdavi-Amiri, *Two new conjugate gradient methods based on modified secant equations*, J. Comput. Appl. Math. **234** (2010), no. 5, 1374–1386.

[7] J. Bai, W. W. Hager and H. Zhang, *An inexact accelerated stochastic ADMM for separable convex optimization*, Comput. Optim. Appl. **81** (2022), no. 2, 479–518.

[8] Y. Chen, M. Cao and Y. Yang, *A new accelerated conjugate gradient method for large-scale unconstrained optimization*, J. Inequal. Appl. **2019**, Paper No. 300, 13 pp.

[9] Y. Chen and Y. Yang, *A three-term conjugate gradient algorithm using subspace for large-scale unconstrained optimization*, Commum. Math. Sci. **18** (2020), no. 5, 1179–1190.

[10] Y.-H. Dai and C.-X. Kou, *A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search*, SIAM J. Optim. **23** (2013), no. 1, 296–320.

[11] Y.-H. Dai and L.-Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim. **43** (2001), no. 1, 87–101.

[12] M. Dawahdeh, M. Mamat, M. Rivaie and S. M. Ibrahim, *Application of conjugate gradient method for solution of regression models*, Int. J. Adv. Sci. Technol. **29** (2020), no. 7, 1754–1763.

[13] S. Devila, M. Malik and W. Giyarti, *A new hybrid PRP-MMSIS conjugate gradient method and its application in portofolio selection*, J. Ris. Aplikasi Mat. **5** (2021), no. 1, 47–59.

[14] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. **91** (2002), no. 2, Ser. A, 201–213.

[15] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim. **2** (1992), no. 1, 21–42.

[16] N. I. M. Gould, D. Orban and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Softw. **29** (2003), no. 4, 373–394.

[17] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim. **16** (2005), no. 1, 170–192.

[18] W. Hu, J. Wu and G. Yuan, *Some modified Hestenes–Stiefel conjugate gradient algorithms with application in image restoration*, Appl. Numer. Math. **158** (2020), 360–376.

[19] A. H. Ibrahim, J. Deepho, A. B. Abubakar and A. Adamu, *A three-term Polak–Ribière–Polyak derivative-free method and its application to image restoration*, Sci. Afr. **13** (2021), e00880, 16 pp.

[20] I. E. Livieris, V. Tampakas and P. Pintelas, *A descent hybrid conjugate gradient method based on the memoryless BFGS update*, Numer. Algorithms **79** (2018), no. 4, 1169–1185.

[21] A. Perry, *A modified conjugate gradient algorithm*, Oper. Res. **26** (1978), no. 6, 1073–1078.

[22] H. Sakai and H. Liduka, *Sufficient descent Riemannian conjugate gradient methods*, J. Optim. Theory Appl. **190** (2021), no. 1, 130–150.

[23] L. Wang, M. Cao, F. Xing and Y. Yang, *The new spectral conjugate gradient method for large-scale unconstrained optimisation*, J. Inequal. Appl. **2020**, Paper No. 111, 11 pp.

[24] C. Wu, J. Wang, J. H. Alcantara and J.-S. Chen, *Smoothing strategy along with conjugate gradient algorithm for signal reconstruction*, J. Sci. Comput. **87** (2021), no. 1, Paper No. 21, 18 pp.

[25] C. Wu, J. Zhan, Y. Lu and J.-S. Chen, *Signal reconstruction by conjugate gradient algorithm based on smoothing l_1-norm*, Calcolo **56** (2019), no. 4, Paper No. 42, 26 pp.

[26] H. Yabe and M. Takano, *Global convergence properties of nonlinear conjugate gradient methods with modified secant condition*, Comput. Optim. Appl. **28** (2004), no. 2, 203–225.

[27] S. Yao and L. Ning, *An adaptive three-term conjugate gradient method based on self-scaling memoryless BFGS matrix*, J. Comput. Appl. Math. **332** (2018), 72–85.

[28] Y.-X. Yuan and J. Stoer, *A subspace study on conjugate gradient algorithms*, Z. Angew. Math. Mech. **75** (1995), no. 1, 69–77.

[29] K. Zhang, H. Liu and Z. Liu, *A new Dai–Liao conjugate gradient method with optimal parameter choice*, Numer. Funct. Anal. Optim. **40** (2019), no. 2, 194–215.

[30] W. Zhou and L. Zhang, *A nonlinear conjugate gradient method based on the MBFGS secant condition*, Optim. Methods Softw. **21** (2006), no. 5, 707–714.

Yueting Yang
School of Mathematics and Statistics, Beihua University, Jilin, Jilin, 132013, China
and
School of Science, Jiangsu University, Zhenjiang, Jiangsu, 212013, China
E-mail address: `yyt2858@163.com`

Xue Zhang, Guoling Zhou and Mingyuan Cao
School of Mathematics and Statistics, Beihua University, Jilin, Jilin, 132013, China
E-mail addresses: `17037385@qq.com`, `zgl9612@163.com`, `cmy0918@beihua.edu.cn`