Optimization: A Journal of Mathematical Programming and Operations Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gopt20

An improved three-term conjugate gradient algorithm for solving unconstrained optimization problems

Songhai Deng[a] & Zhong Wan[a]
[a] School of Mathematics and Statistics, Central South University, Changsha, China.

To cite this article: Songhai Deng & Zhong Wan (2014): An improved three-term conjugate gradient algorithm for solving unconstrained optimization problems, Optimization: A Journal of Mathematical Programming and Operations Research, DOI: 10.1080/02331934.2014.984706

To link to this article: http://dx.doi.org/10.1080/02331934.2014.984706

Optimization, 2014
http://dx.doi.org/10.1080/02331934.2014.984706

Taylor & Francis
Taylor & Francis Group

An improved three-term conjugate gradient algorithm for solving unconstrained optimization problems

Songhai Deng and Zhong Wan*

School of Mathematics and Statistics, Central South University, Changsha, China

(*Received 1 March 2014; accepted 21 October 2014*)

In this article, we present an improved three-term conjugate gradient algorithm for large-scale unconstrained optimization. The search directions in the developed algorithm are proved to satisfy an approximate secant equation as well as the Dai-Liao's conjugacy condition. With the standard Wolfe line search and the restart strategy, global convergence of the algorithm is established under mild conditions. By implementing the algorithm to solve 75 benchmark test problems with dimensions from 1000 to 10,000, the obtained numerical results indicate that the algorithm outperforms the state-of-the-art algorithms available in the literature. It costs less CPU time and smaller number of iterations in solving the large-scale unconstrained optimization.

Keywords: algorithms; optimization; conjugate gradient method; global convergence

AMS Subject Classifications: 90C30; 62K05

1. Introduction

Consider the following problem:

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{1}$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable such that its gradient is available. Let $g: \mathbb{R}^n \to \mathbb{R}^n$ denotes the gradient function of f, and let g_k denotes the value of g at x_k.

Let $x_0 \in \mathbb{R}^n$ be an initial point. A sequence of approximate solutions x_k of (1) is often generated by

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

where $k \geq 0$, α_k is a stepsize obtained by some line search rule and d_k is a search direction (see [1,2]). In the classical conjugate gradient methods, d_k is given by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k > 0. \end{cases} \tag{3}$$

In (3), β_k is called the conjugate parameter. With a different choice of β_k, the obtained method has distinct numerical performance.

---

*Corresponding author. Email: wanmath@csu.edu.cn

Denote $y_k = g_{k+1} - g_k$. We present some popular conjugate parameters as follows:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \text{ (Hestences and Stiefel [3]),}$$

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \text{ (Fletcher and Reeves [4]),}$$

$$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k} \text{ (Dai and Yuan [5]),}$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

$$\beta_k^{HZ} = \frac{1}{d_k^T y_k}\left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k}\right) g_{k+1} \text{ (Hager and Zhang [6]).}$$

Recently, a type of three-term conjugate gradient methods has been widely studied. The first general three-term conjugate gradient method was proposed in [7] by Beale, and the search direction is given by

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \gamma_k d_t, \tag{4}$$

where $\beta_k = \beta_k^{HS}$ (or β_k^{FR} and β_k^{DY}, etc.),

$$\gamma_k = \begin{cases} 0, & k = t+1, \\ \dfrac{g_{k+1}^T y_t}{d_t^T y_t}, & k > t+1, \end{cases} \tag{5}$$

and d_t is a restart direction. Subsequently, in [8], another three-term conjugate gradient method was proposed, where the search direction was specified by:

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_{k-1}} d_{k-1}, \tag{6}$$

with $d_{-1} = 0, d_0 = -g_0$. It has been proved that for a convex quadratic function f, the search directions generated by (6) are conjugate with respect to the coefficients matrix of the second-order term. As an extension of the classic conjugate gradient method, the following two three-term conjugate gradient methods:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{g_k^T g_k} d_k - \frac{g_{k+1}^T d_k}{g_k^T g_k} y_k, \tag{7}$$

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{s_k^T y_k} s_k - \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k. \tag{8}$$

were presented in [9,10], respectively. (7) is called the descent modified PRP conjugate gradient algorithm, and (8) is referred to as the descent modified HS conjugate gradient method. A notable property of these methods is that the search directions can be proved to be sufficiently descent. Actually, d_k in [9] or [10] satisfies $g_k^T d_k = -\|g_k\|^2$ for each $k \geq 0$.

Very recently, a series of three descent conjugate gradient algorithms were developed by Andrei in [11–13]. Let $s_k = x_{k+1} - x_k$. The search directions in these algorithms are determined by

$$d_{k+1} = -\frac{y_k^T s_k}{\|g_k\|^2} g_{k+1} + \frac{y_k^T g_{k+1}}{\|g_k\|^2} s_k - \frac{s_k^T g_{k+1}}{\|g_k\|^2} y_k, \tag{9}$$

$$d_{k+1} = -g_{k+1} - \left(\left(1 + \frac{\|y_k\|^2}{y_k^T s_k}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k}\right) s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k, \tag{10}$$

and

$$d_{k+1} = -g_{k+1} - \left(\left(1 + 2\frac{\|y_k\|^2}{y_k^T s_k}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k}\right) s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k, \tag{11}$$

respectively. Different from (7) and (8), the directions in (9)–(11) satisfy the sufficiently descent condition as well as the following extended conjugacy condition in [14]:

$$d_{k+1}^T y_k = -t_k g_{k+1}^T s_k, \tag{12}$$

where $t_k > 0$. Numerical experiments showed that the three-term conjugate gradient method in [12] outperforms the other six three-term algorithms in [9–11,15–17].

In this article, we intend to construct a new search direction in the framework of three-term conjugate gradient method. This direction is required to satisfy the Dai-Liao's conjugacy condition (12) as well as an approximate secant equation (*not being sufficiently descent*). Since the search direction satisfies an approximate secant condition and is closely related with the limited memory BFGS method, it can greatly improve the numerical performance of the classic conjugate gradient method. On the other hand, different from the quasi-Newton method, the constructed direction in this article does not need to compute or store any approximate Hessian matrix of the objective function. Thus, the cost of computation at each iteration of algorithm is similar to that in the conjugate gradient method.

In addition, different from the ordinary restart strategy, the developed algorithm in this article may restart with an associated two-term conjugate gradient method (*versus a standard conjugate gradient method*) if necessary.

The rest of this article is organized as follows. In next section, we will state the idea to propose a new spectral conjugate gradient method in detail, and then a new algorithm is developed. Global convergence is established in Section 3. Section 4 is devoted to numerical experiments. Some conclusions are drawn in the last section.

2. Algorithm: improved three-term conjugate gradient (ITTCG)

In this section, we will construct a new three-term conjugate gradient direction. Then, an algorithm will be developed.

Let d_{k+1} be the search direction at the $(k+1)$th iteration. If d_{k+1} is generated by the quasi-Newton method, then it should be required to satisfy the following condition (see [18–22]):

$$s_k^T \nabla^2 f(x_{k+1}) d_{k+1} = -s_k^T g_{k+1}. \tag{13}$$

Since it is difficult to compute and store the Hessian matrix $\nabla^2 f(x_{k+1})$ in solving large-scale optimization problems, we replace (13) by

$$y_k^T d_{k+1} = -s_k^T g_{k+1}, \tag{14}$$

where the secant condition $\nabla^2 f_{k+1} s_k = y_k$ is employed.

To obtain a direction d_{k+1} satisfying (14), we construct d_{k+1} as follows:

$$d_{k+1} = -g_{k+1} - \left(\left(1 + \frac{\|y_k\|^2}{y_k^T s_k} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \right) s_k + \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k. \tag{15}$$

It is noted that the main difference between (15) and (10) or (11) lies in the sign of the last term. As in (15), the positive sign ensures that d_{k+1} satisfies the condition (14). However, for the directions given by (10) and (11), the equality (14) does not hold.

Unfortunately, the search direction in (15) is not always descent as a quasi-Newton direction. For this reason, we use the restart strategy to develop a well-defined algorithm. Similar to the Beale's three-term method, we restart the direction with a two-term conjugate gradient direction if it is descent. If the two-term conjugate gradient direction is still ascent, then the algorithm restarts with the steepest descent direction $-g_{k+1}$. Specifically, we choose a search direction as follows:

$$d_{k+1} = -g_{k+1} - \delta_k s_k + \eta_k y_k, \tag{16}$$

where

$$\delta_k = \begin{cases} \left(1 + \frac{\|y_k\|^2}{y_k^T s_k} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \triangleq \bar{\delta}_k, & \text{if } (-g_{k+1} - \bar{\delta}_k s_k)^T g_{k+1} < 0, \\ 0, \quad \text{otherwise}, \end{cases} \tag{17}$$

and

$$\eta_k = \begin{cases} \frac{s_k^T g_{k+1}}{y_k^T s_k}, & \text{if } g_{k+1}^T s_k g_{k+1}^T y_k < 0, \\ 0, & \text{otherwise}. \end{cases} \tag{18}$$

Along the obtained direction given by (16)–(18), we find a suitable step length by the standard Wolfe line search

$$\begin{cases} f(x_{k+1}) \le f(x_k) + \delta g(x_k)^T d(x_k) \\ g(x_{k+1})^T d(x_k) \ge \sigma g(x_k)^T d(x_k), \end{cases} \tag{19}$$

or the strong Wolfe line search rule (see [15,23–25]) :

$$\begin{cases} f(x_{k+1}) \le f(x_k) + \delta g(x_k)^T d(x_k) \\ |g(x_{k+1})^T d(x_k)| \le \sigma |g(x_k)^T d(x_k)|. \end{cases} \tag{20}$$

With the above preparation, we now in a position to state the overall framework of our algorithm.

Algorithm 2.1 (ITTCG) *Step 1* Take constants δ and σ, $0 < \delta \le \sigma < 1$. Choose an initial point $x_0 \in dom f$ and compute $f_0 = f(x_0)$ and $g_0 = \nabla g(x_0)$. Set $k := 0$.

Step 2 If $\|g_k\|_\infty < \epsilon$, then the algorithm stops. Otherwise, go to Step 3.

Step 3 Determine the stepsize α_k by the Wolfe line search conditions (19)

Step 4 Update the iterate point as $x_{k+1} := x_k + \alpha_k d_k$. Compute f_{k+1}, g_{k+1}, $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.

Step 5 Determine δ_k and η_k by (17) and (18), respectively.

Step 6 Compute the search direction by (16).

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

3. Global convergence

In this section, we are going to study the global convergence of Algorithm 2.1.

We first state the following mild assumptions, which are required to prove the main results in this article.

ASSUMPTION 3.1 *The level set* $\Omega = \{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\}$ *is bounded.*

ASSUMPTION 3.2 *In some neighbourhood* N *of* Ω, f *is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant* $L > 0$ *such that*

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall x, y \in N. \tag{21}$$

Remark 3.1 Since $f(x_k)$ is decreasing, it is clear that the sequence $\{x_k\}$ generated by Algorithm 2.1 is contained in a bounded region from Assumption 3.1. So, there exists a convergent subsequence of $\{x_k\}$. Without loss of generality, it is supposed that $\{x_k\}$ is convergent. In addition, from Assumptions 3.1 and 3.2, it follows that there is a constant $\gamma > 0$ such that $\|g(x)\| \le \gamma, \forall x \in \Omega$. Thus, the sequence $\{g_k\}$ is bounded.

We first prove the following results.

LEMMA 3.1 *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuously differentiable function. Then, the direction generated by Algorithm* 2.1 *is descent.*

Proof From the Wolfe conditions (19), it follows that $y_k^T s_k > 0$. Thus, if $(g_{k+1}^T s_k)(g_{k+1}^T y_k) < 0$, then

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \left(1 + \frac{\|y_k\|^2}{y_k^T s_k}\right)\frac{(g_{k+1}^T s_k)^2}{y_k^T s_k} + 2\frac{g_{k+1}^T s_k g_{k+1}^T y_k}{y_k^T s_k} < 0.$$

Otherwise, we have $\eta_k = 0$. Hence,

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \left(1 + \frac{\|y_k\|^2}{y_k^T s_k}\right)\frac{(g_{k+1}^T s_k)^2}{y_k^T s_k} + \frac{g_{k+1}^T s_k g_{k+1}^T y_k}{y_k^T s_k}.$$

If $g_{k+1}^T d_{k+1} < 0$, d_{k+1} reduce to a direction given by the associated two-term conjugate gradient method. Otherwise, $d_{k+1} = -g_{k+1}$ and $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 < 0$.

Based on the above discussion, we conclude that the direction generated by Algorithm 2.1 is descent. □

LEMMA 3.2 *Suppose that the stepsize* α_k *satisfies the Wolfe conditions* (19). *Then,* d_{k+1} *given by* (16) *satisfies the conjugacy condition* (12) *apart from the restarted search directions.*

Proof Since $g_{k+1}^T s_k g_{k+1}^T y_k < 0$, it follows from (16) that

$$y_k^T d_{k+1} = -s_k^T g_{k+1}.$$

Thus, (12) holds with $t_k = 1 > 0$.

If $g_{k+1}^T s_k g_{k+1}^T y_k \geq 0$, then by direct calculation,

$$y_k^T d_{k+1} = -\left(1 + \frac{\|y_k\|^2}{y_k^T s_k}\right) s_k g_{k+1}.$$

Thus, (12) holds with $t_k = 1 + \dfrac{\|y_k\|^2}{y_k^T s_k} > 0$. □

Remark 3.2 In the memoryless version of BFGS algorithm, the inverse approximate Hessian matrix of f is updated by

$$H_{k+1} = H_k - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k} + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right) \frac{s_k s_k^T}{y_k^T s_k}. \tag{22}$$

If we set $H_k = I$ in (22), then d_{k+1} in (16) satisfies $d_{k+1} = -H_{k+1} g_{k+1}$. Thus, our search direction is an approximate quasi-Newton direction.

LEMMA 3.3 *Suppose that d_k is a descent direction and that the gradient ∇f satisfies the Lipschitz condition:*

$$\|\nabla f(x) - \nabla f(x_k)\| \leq L\|x - x_k\|$$

for all $x \in [x_k, x_{k+1}]$. Under the Wolfe line search, it holds that

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

Proof Subtracting $g_k^T d_k$ from the both sides of the first inequality in (19), we obtain

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

Since d_k is descent and $0 < \sigma < 1$, the conclusion follows immediately. □

LEMMA 3.4 *Let $\{d_k\}$ and $\{\alpha_k\}$ be two sequences generated by Algorithm 2.1. Suppose that Assumptions (3.1) and (3.2) hold. Then,*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{23}$$

Proof From (19) and Lemma 3.3, it follows that

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

In virtue of Assumption 3.1, we get the Zoutendijk condition (23). □

Powell [26] gives the following property that in conjugate gradient algorithm the iteration can fail, in the sense that $\|g_k\| \geq \gamma_1 > 0$ for all k, only if $\|d_k\| \to \infty$ sufficiently rapid. More precise, the sequence of gradient norms $\|g_k\|$ can be bounded away from zero only if $\sum_{k \geq 0} 1/\|d_k\| < \infty$. For any conjugate gradient method with strong Wolfe line search (20) the following general results holds.[27]

Proposition 3.1 *Let $\{d_k\}$ and $\{\alpha_k\}$ be two sequences generated by Algorithm 2.1. Suppose that Assumptions 3.1 and 3.2 hold. For any conjugate gradient method (2), if*

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=+\infty \tag{24}$$

holds then

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

Theorem 3.1 *Let $\{x_k\}$ be a sequence generated by Algorithm 2.1. Suppose that Assumptions 3.1 and 3.2 hold. If f is a uniformly convex function on the level set Ω, i.e. there exists a constant $\mu > 0$ such that*

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq \mu\|x-y\|^2$$

for all $x,\ y$ in the neighbourhood N of Ω, then,

$$\lim_{k\to\infty}\|g_k\|=0. \tag{25}$$

Proof By Assumptions 3.1 and 3.2, we know that $\|g_k\| < \gamma$ and $\|y_k\|\leq L\|s_k\|$. Since f is a uniformly convex function on the level set Ω, it is clear that

$$y_k^Ts_k\geq \mu\|s_k\|^2.$$

Thus,

$$\begin{aligned}|\delta_k| &\leq \frac{|s_k^Tg_{k+1}|}{|y_k^Ts_k|}+\frac{\|y_k\|^2|s_k^Tg_{k+1}|}{|y_k^Ts_k|^2}+\frac{|y_k^Tg_{k+1}|}{|y_k^Ts_k|}\\ &\leq \frac{\gamma}{\mu\|s_k\|}+\frac{L^2\gamma}{\mu^2\|s_k\|}+\frac{L\gamma}{\mu\|s_k\|} \qquad (26)\\ &=\frac{\gamma}{\mu}\left(1+L+\frac{L^2}{\mu}\right)\frac{1}{\|s_k\|},\end{aligned}$$

and

$$|\eta_k|\leq \frac{|s_k^Tg_{k+1}|}{|y_k^Ts_k|}\leq \frac{\|s_k\|\|g_{k+1}\|}{\mu\|s_k\|^2}\leq \frac{\gamma}{\mu\|s_k\|}. \tag{27}$$

From (26) and (27), it follows that

$$\|d_{k+1}\|\leq \|g_{k+1}\|+|\delta_k|\|s_k\|+|\eta_k|\|y_k\|\leq \gamma+\frac{\gamma}{\mu}\left(2+L+\frac{L^2}{\mu}\right).$$

Therefore,

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=+\infty.$$

In view of Proposition 3.1, we know that

$$\liminf_{k\to\infty}\|g_k\|=0. \tag{28}$$

For uniformly convex functions it is equivalent to (25). □

THEOREM 3.2 *Let* $\{x_k\}$ *be a sequence generated by Algorithm 2.1. Suppose that Assumptions 3.1 and 3.2 hold. If there exists a constant* $c > 0$, *such that* $y_k^T s_k \geq c$ *for* k *large enough. Then,*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{29}$$

Proof Since $g_k^T s_k < 0$ for any k, we know

$$s_k^T g_{k+1} = y_k^T s_k + g_k^T s_k < y_k^T s_k.$$

By Assumptions 3.2 and 3.1, it is clear that

$$\|y_k\| = \|g_{k+1} - g_k\| \leq L\|s_k\|,$$

and there exists a constant $B > 0$ such that $\forall x \in \Omega$ (the level set)

$$\|x - x_0\| \leq B.$$

Thus, $\forall x', x'' \in \Omega$,

$$\|x' - x''\| \leq \|x' - x_0\| + \|x'' - x_0\| \leq 2B.$$

It implies that

$$\|s_k\| = \|x_{k+1} - x_k\| \leq 2B, \quad \|y_k\| \leq 2BL.$$

Therefore, for k large enough,

$$\begin{aligned} |\delta_k| &\leq \left(1 + \frac{\|y_k\|^2}{|y_k^T s_k|}\right) \frac{|s_k^T g_{k+1}|}{|y_k^T s_k|} + \frac{|y_k^T g_{k+1}|}{|y_k^T s_k|} \\ &< 1 + \frac{\|y_k\|^2}{|y_k^T s_k|} + \frac{|y_k^T g_{k+1}|}{|y_k^T s_k|} \\ &\leq 1 + \frac{4B^2L^2}{c} + \frac{2BL\gamma}{c} \equiv M_1, \end{aligned} \tag{30}$$

and

$$|\eta_k| < \frac{|s_k^T g_{k+1}|}{y_k^T s_k} \leq \frac{\|s_k\|\|g_{k+1}\|}{y_k^T s_k} < \frac{2B\gamma}{c} \equiv M_2. \tag{31}$$

Consequently, we have

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\delta_k|\|s_k\| + |\eta_k|\|y_k\| \leq \gamma + 2BM_1 + 2BLM_2.$$

Similar to the proof of Theorem 3.1, we conclude that

$$\sum_{k\geq 1} \frac{1}{\|d_k\|^2} = +\infty.$$

By property 3.1, (29) holds. □

Remark 3.3 It is noted that in the establishment of global convergence, we suppose that $y_k^T s_k \geq c$ for k large enough, where $c > 0$ is a positive constant. This condition is also required in [12,13] to prove the main results (see Theorem 5.2 in [12]). Since, the search direction d_k is descent and the stepsize α_k is found by the Wolfe line search, it is clear that for all $k \geq 0$, the inequality $y_k^T s_k > 0$ holds. From this point of view, the condition $y_k^T s_k \geq c$ is a bit stronger. However, as for a uniformly convex objective function, the global convergence has been proved without this assumption (see 3.1). In the implementation of Algorithm 2.1 (see Section 4), we take $c = 10^{-30}$, as done in [12,13]. In other words, if $y_k^T s_k > 10^{-30}$, the three-term method is employed to choose the search direction. Otherwise, d_k is obtained by the associated two-term method or the steepest descent method.

4. Numerical experiments

In this section, we shall report the numerical performance of Algorithm 2.1.

We test Algorithm 2.1 (ITTCG) by implementing it to solve the 75 benchmark test problems from [28]. For each test problem, we undertake 10 experiments with the dimension of the problem increasing as $n = 1000, 2000, \ldots, 10,000$. In the Wolfe line search, the parameters $\delta = 0.0001$, $\rho = 0.8$ and $\sigma = 0.8$. The tolerance of the algorithms takes 10^{-6}.

We compare the ITTCG with the TTCG, developed by Andrei in [13], the THREECG in [12] and the CG_DESCENT by Hager and Zhang in [29], which have been reported to be very efficient for solving the non-convex unconstrained optimization problems. In these algorithms, the Wolfe line search strategy is employed.

All codes of the computer procedures are written in Fortran 77, and are implemented on PC with 2.9 GHz CPU processor, 4 GB RAM memory and Windows XP operation system. The code of CG_DESCENT is downloaded from the web site:

http://www.math.ufl.edu/~~hager/papers/CG/cg_descent.f

The code of TTCG algorithm and the Wolfe Line search is downloaded from N. Andrei's web site:

http://camo.ici.ro/neculai/THREECG/threecg.for

The code of the Wolfe Line search is also used in the THREECG.

For all the algorithms apart from the CG_DESCENT, we fix the maximal number of iteration, specified by maxiter $(= 10,000)$. The maximal number of function and gradient evaluation is denoted by maxifg$(= 15,000)$. If an algorithm exceeds one of the maximal values in solving a test problem, then it is regarded as failure in solving the problem. For the CG_DESCENT, we adopt its default maximum values.

In the numerical experiments, it is revealed that Algorithm 2.1 (ITTCG) does not often restart. Out of the 750 test problems, there are only four problems (less than 1%) where the search direction of algorithm 2.1 restarts with $-g_{k+1}$ as it is ascending. This implies that the search direction defined by (15) is often efficient to search for the minimizer of problem (1). In addition, if the constant c takes 10^{-30} in Theorem 3.2, $y_k^T s_k < c$ appears only for one of the 75 test problems (Problem 71).

By the profiles of Dolan and Moré in [30], Figure 1 shows the CPU time performances of Algorithm 2.1 (ITTCG), the TTCG, the THREECG and the CG_DESCENT, respectively. Specifically, in the CPU time profiles, we plot the percentage P of problems for which the method is within a time τ for each algorithm. In other words, the curves in Figure 1 indicate the percentage of the test problems for which an algorithm has solved in a given time. Thus, the obtained CPU time profiles can intuitively describe the overall performance of the algorithms.

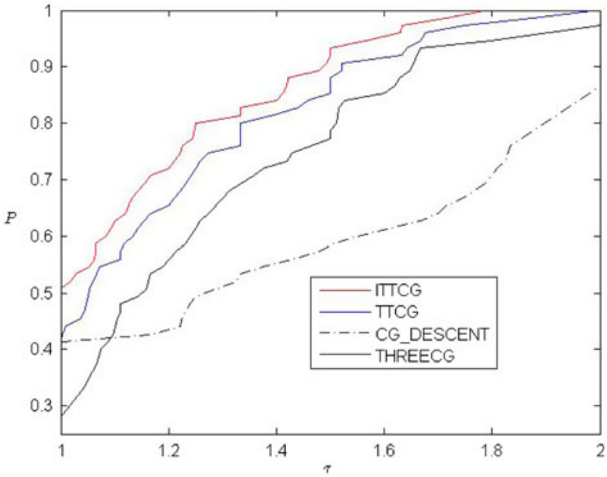

Figure 1. Performance profile of CPU time.

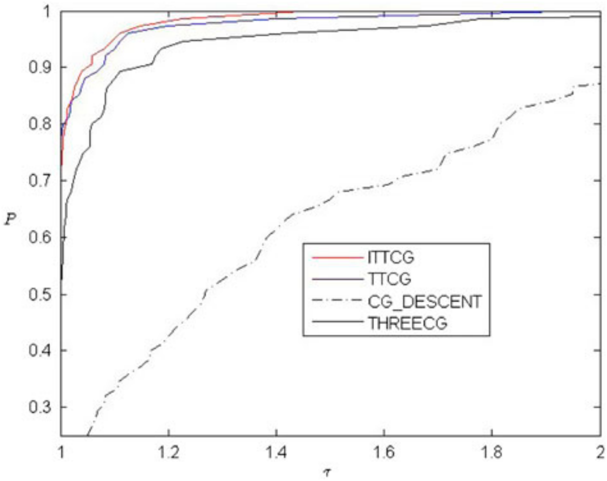

Figure 2. Performance profile of iteration.

Since the CPU time is often affected by the environment of computer, such as the busy or free task status and the operation system, we further make a comparison among the four algorithms by the number of iteration and the number of evaluation of the function and the gradient. These numbers are often fixed for an algorithm to solve the same test problem (see [30]).

In Figure 2, we report the iteration performance of Algorithm 2.1 (ITTCG), the TTCG, the THREECG and the CG_DESCENT, respectively. Figure 3 is used to report the function and the gradient evaluation performance of Algorithm 2.1 (ITTCG), the TTCG, the THREECG and the CG_DESCENT, respectively.

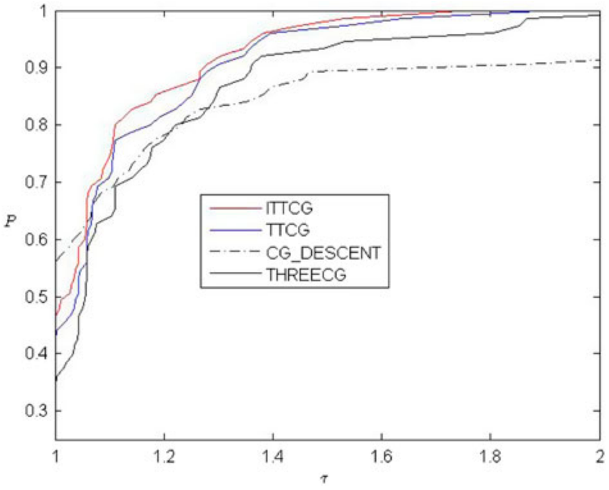

Figure 3. Performance profile of evaluation number for functions and gradients.

From Figure 1, out of the 750 test problems, there are 660 ones where the occupation of CPU time by the ITTCG is less than that by the TTCG and the THREECG, respectively. Figure 2 demonstrates that the ITTCG performs slightly better than the TTCG, but outperforms the CG_DESCENT for all of the 750 test problems. Figure 3 reveals that the numerical performances of the algorithms are similar as for the number of evaluation of the function and the gradient. However, the ITTCG seems to perform relatively better.

Taken together, the obtained numerical results indicate that the ITTCG performs better than the other two three-term conjugate gradient algorithms (TTCG and THREECG), and all of the three-term methods outperform the two-term CG_DESCENT method for the 750 test problems in this article.

5. Conclusion

In this article, we have proposed an improved three-term conjugate gradient method for solving non-linear large-scale unconstrained optimization problems. The search directions of the developed algorithm have been proved to satisfy an approximate secant equation as well as the Dai-Liao's conjugacy condition. With suitable assumptions, global convergence of the algorithm has been established.

Numerical experiments have shown the efficiency of the algorithm in solving many large-scale benchmark test problems. Compared with the state-of-the-art algorithms available in the literature, our algorithm costs less CPU time and smaller number of iteration in finding the optimal solution.

Funding

This work is supported by Natural Science Foundation of Hunan Province [14JJ2003, 13JJ3002] and National Natural Science Foundation of China [grant number 71210003], [grant number 71221061].

References

[1] Wan Z, Hu CM, Yang ZL. A spectral PRP conjugate gradient methods for nonconvex optimization problem based on modified line search. Discrete Cont. Dyn. Syst-Ser. B. 2011;16:1153–1169.

[2] Wan Z, Yang ZL, Wang YL. New spectral PRP conjugate gradient method for unconstrained optimization. Appl. Math. Lett. 2011;24:16–22.

[3] Hestenes MR, Stiefel EL. Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bureau Stand. 1952;49:409–436.

[4] Fletcher R, Reeves CM. Function minimization by conjugate gradients. Comput. J. 1964;7:149–154.

[5] Dai YH, Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. 1999;10:177–182.

[6] Hager WW, Zhang H. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 2005;16:170–192.

[7] Beale EML. A derivative of conjugate gradients. In: Lootsma FA, editor. Numerical methods for nonlinear optimization. London: Academic Press; 1972. p. 39–43.

[8] Nazareth L. A conjugate direction algorithm without line search. J. Optim. Theory Appl. 1997;23:373–387.

[9] Zhang L, Zhou W, Li DH. A descent modified Polak–Ribiére–Polyak conjugate gradient method and its global convergence. IMA J. Numer. Anal. 2006;26:629–640.

[10] Zhang L, Zhou W, Li DH. Some descent three-term conjugate gradient methods and their global convergence. Optim. Meth. Softw. 2007;22:697–711.

[11] Andrei N. A modified Polak–Ribière–Polyak conjugate gradient algorithm for unconstrained optimization. Optimization. 2011;60:1457–1471.

[12] Andrei N. A simple three-term conjugate gradient algorithm for unconstrained optimization. J. Comput. Appl. Math. 2013;241:19–29.

[13] Andrei N. On three-term conjugate gradient algorithms for unconstrained optimization. Appl. Math. Comput. 2013;219:6316–6327.

[14] Dai YH, Liao LZ. New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. 2001;43:87–101.

[15] Al-Bayati Abbas Y, Sharif WH. A new three-term conjugate gradient method for unconstrained optimization. Can. J. Sci. Eng. Math. 2010;1:108–124.

[16] Cheng W. A two-term PRP-based descent method. Numer. Funct. Anal. Optim. 2007;28:1217–1230.

[17] Zhang J, Xiao Y, Wei ZX. Nonlinear conjugate gradient methods with sufficient descent condition for large-scale unconstrained optimization. Math. Probl. Eng. 2009;2009. doi:10.1155/2009/234290.

[18] Andrei N. New accelerated conjugate gradient algorithms as a modification of Dai-Yuan's computational scheme for unconstrained optimization. J. Comput. Appl. Math. 2010;234:3397–3410.

[19] Deng SH, Wan Z, Chen XH. An improved spectral conjugate gradient algorithm for nonconvex unconstrained optimization problems. J. Optim. Theory Appl. 2013;157:820–842.

[20] Wan Z, Chen Y, Huang S, Feng DD. A modified nonmonotone BFGS algorithm for solving smooth nonlinear equations. Optim. Lett. 2014;8:1845–1860.

[21] Wan Z, Huang S, Zheng XD. New cautious BFGS algorithm based on modified Armijo-type line search. J. Inequ. Appl. 2012;2012:1–10. doi:10.1186/1029-242X-2012-241.

[22] Wan Z, Teo KL, Shen XL, Hu CM. New BFGS method for unconstrained optimization problem based on modified Armijo line search. Optimization. 2014;63:285–304.

[23] Andrei N. Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions. J. Optim. Theory Appl. 2013;159:159–182.

[24] Dai YH, Yuan Y. A three-parameter family of nonlinear conjugate gradient methods. Math. Comput. 2001;70:1155–1167.

[25] Li C, Fang L, Cao XL. Global convergence of a kind of conjugate gradient method. TELKOMNIKA. 2013;11:544–549.

[26] Powell MJD. Nonconvex minimization calculations and the conjugate gradient method. In: Griffiths DF, editor. Numerical analysis. Berlin: Springer; 1984. p. 122–141.

[27] Nocedal J. Conjugate gradient methods and nonlinear optimization. In: Adams L, Nazareth JL, editors. Linear and nonliear conjugate gradient related methods. Philadelphia, PA: SIAM; 1996. p. 9–23.

[28] Andrei N. An unconstrained optimization test functions collection. Adv. Model. Optim. 2008;10:147–161.

[29] Hager WW, Zhang H. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. ACM Trans. Math. Softw. 2006;32:113–137.

[30] Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Math. Program. 2002;91:201–213.

[31] Huang S, Wan Z, Deng SH. A modified projected conjugate gradient method for unconstrained optimization problems. ANZIAM J. 2013;54:143–152.