Mathematical Sciences (2020) 14:1–9
https://doi.org/10.1007/s40096-019-00310-y

ORIGINAL RESEARCH

A globally convergent hybrid conjugate gradient method with strong Wolfe conditions for unconstrained optimization

P. Kaelo[1] · P. Mtagulwa[1] · M. V. Thuto[1]

Received: 25 September 2019 / Accepted: 24 October 2019 / Published online: 5 November 2019

Abstract

In this paper, we develop a new hybrid conjugate gradient method that inherits the features of the Liu and Storey (LS), Hestenes and Stiefel (HS), Dai and Yuan (DY) and Conjugate Descent (CD) conjugate gradient methods. The new method generates a descent direction independently of any line search and possesses good convergence properties under the strong Wolfe line search conditions. Numerical results show that the proposed method is robust and efficient.

Keywords Unconstrained optimization · Global convergence · Sufficient descent · Strong Wolfe conditions

Mathematics Subject Classification 90C30 · 65K05 · 90C06

Introduction

In this paper, we consider solving the unconstrained optimization problem

$$\min f(x), \tag{1}$$

where $x \in \mathbb{R}^n$ is an n-dimensional real vector and $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function, using a nonlinear conjugate gradient method. Optimization problems arise naturally in problems from many scientific and operational applications (see e.g. [12, 19–22, 35, 36], among others).

To solve problem (1), a nonlinear conjugate gradient method starts with an initial guess, $x_0 \in \mathbb{R}^n$, and generates a sequence $\{x_k\}_{k=0}^{\infty}$ using the recurrence

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

where the step size α_k is a positive parameter and d_k is the search direction defined by

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

The scalar β_k is the conjugate gradient update coefficient and $g_k = \nabla f(x_k)$ is the gradient of f at x_k. In finding the step size α_k, the inexpensive line searches such as the weak Wolfe line search

$$\begin{cases} f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^T d_k \\ g_{k+1}^T d_k \ge \sigma g_k^T d_k, \end{cases} \tag{4}$$

the strong Wolfe line search

$$\begin{cases} f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^T d_k \\ |g_{k+1}^T d_k| \le \sigma |g_k^T d_k|, \end{cases} \tag{5}$$

or the generalized Wolfe conditions

$$\begin{cases} f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^T d_k \\ \sigma g_k^T d_k \le g_{k+1}^T d_k \le -\sigma_1 g_k^T d_k, \end{cases} \tag{6}$$

where $0 < \delta < \sigma < 1$ and $\sigma_1 \ge 0$ are constants, are often used. Generally, conjugate gradient methods differ by the choice of the coefficient β_k. Well-known formulas for β_k can be divided into two categories. The first category includes Fletcher and Reeves (FR) [11], Dai and Yuan (DY) [6] and Conjugate Descent (CD) [10]:

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}},$$

where $\|\cdot\|$ denotes the Euclidean norm and $y_{k-1} = g_k - g_{k-1}$. These methods have strong convergence properties. However, since they are very often susceptible to jamming, they tend to have poor numerical performance. The other

✉ P. Kaelo
kaelop@mopipi.ub.bw

[1] Department of Mathematics, University of Botswana, Private Bag UB00704, Gaborone, Botswana

category includes Hestenes and Stiefel (HS) [16], Polak-Ribière-Polyak (PRP) [28, 29] and Liu and Storey (LS) [26]:

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \quad \beta_k^{LS} = -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}.$$

Although these methods may fail to converge, they have an in-built automatic restart feature which helps them avoid jamming and hence makes them numerically efficient [5].

In view of the above stated drawbacks and advantages, many researchers have proposed hybrid conjugate gradient methods that combine different β_k coefficients so as to limit the drawbacks and maximize in the advantages of the original respective conjugate gradient methods. For instance, Touati-Ahmed and Storey [31] suggested one of the first hybrid method where the coefficient β_k is given by

$$\beta_k^{TS} = \begin{cases} \beta_k^{PRP}, & \text{if} \quad 0 \le \beta_k^{PRP} \le \beta_k^{FR}, \\ \beta_k^{FR}, & \text{otherwise.} \end{cases}$$

The authors proved that β_k^{TS} has good convergence properties and numerically outperforms both the β_k^{FR} and β_k^{PRP} methods. Alhawarat et al. [3] introduced a hybrid conjugate gradient method in which the conjugate gradient update coefficient is computed as

$$\beta_k^{AZPRP} = \begin{cases} \dfrac{\|g_k\|^2 - g_k^T g_{k-1}}{\|g_{k-1}\|^2}, & \text{if } \|g_k\|^2 > |g_k^T g_{k-1}|, \\ \dfrac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{\|g_{k-1}\|^2}, & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ 0, & \text{otherwise,} \end{cases}$$

where μ_k is defined as

$$\mu_k = \frac{\|x_k - x_{k-1}\|}{\|y_{k-1}\|}.$$

The authors proved that the method possesses global convergence property when weak Wolfe line search is employed. Moreover, numerical results demonstrate that the proposed method outperforms both the CG-Descent 6.8 [14] and CG-Descent 5.3 [13] methods on a number of benchmark test problems.

In [5], Babaie-Kafaki gave a quadratic hybridization of β_k^{FR} and β_k^{PRP}, where

$$\beta_k^{HQ\pm} = \begin{cases} \beta_k^{+}(\theta_k^{\pm}), & \theta_k^{\pm} \in [-1, 1], \\ \max(0, \beta_k^{PRP}), & \theta_k^{\pm} \in \mathbb{C}, \\ -\beta_k^{FR}, & \theta_k^{\pm} < -1, \\ \beta_k^{FR}, & \theta_k^{\pm} > 1, \end{cases}$$

and the hybridization parameter $\theta_k^{\pm}$ is taken from the roots of the quadratic equation

$$\theta_k^2 \beta_k^{PRP} - \theta_k \beta_k^{FR} + \beta_k^{HS} - \beta_k^{PRP} = 0,$$

that is

$$\theta_k^{\pm} = \frac{\beta_k^{FR} \pm \sqrt{(\beta_k^{FR})^2 - 4\beta_k^{PRP}(\beta_k^{HS} - \beta_k^{PRP})}}{2\beta_k^{PRP}},$$

and

$$\beta_k^{+}(\theta_k) = \max(0, \beta_k^{PRP})(1 - \theta_k^2) + \theta_k \beta_k^{FR}, \quad \theta_k \in [-1, 1].$$

Thus, the author suggested two methods β_k^{HQ+} and β_k^{HQ-}, corresponding to θ^+ and θ^-, respectively, with numerical results showing β_k^{HQ-} to be more efficient than β_k^{HQ+}.

More recently, Salih et al. [30] presented another hybrid conjugate gradient method defined by

$$\beta_k^{YHM} = \begin{cases} \dfrac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2}, & \text{if } 0 \le g_k^T g_{k-1} \le \|g_k\|^2, \\ \dfrac{g_k^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1}\right)}{\|g_{k-1}\|^2}, & \text{otherwise.} \end{cases}$$

The authors showed that the β_k^{YHM} method satisfies the sufficient descent condition and possesses global convergence property under the strong Wolfe line search. In 2019, Faramarzi and Amini [9] introduced a spectral conjugate gradient method defined as

$$\beta_k^{ZDK} = \frac{g_k^T z_{k-1}}{d_{k-1}^T z_{k-1}} - \frac{\|z_{k-1}\|^2}{d_{k-1}^T z_{k-1}} \frac{g_k^T d_{k-1}}{d_{k-1}^T z_{k-1}},$$

with the spectral search direction given as

$$d_k = -\theta_k g_k + \beta_k^{ZDK} d_{k-1}, \; d_0 = -g_0,$$

where

$$z_{k-1} = y_{k-1} + h_k ||g_{k-1}||^r s_{k-1}, \; s_{k-1} = x_k - x_{k-1} \text{ and } h_k = D + \max\left\{ -\frac{s_{k-1}^T y_{k-1}}{||s_{k-1}||^2}, 0 \right\} ||g_{k-1}||^{-r},$$

with r and D being positive constants. The authors suggested computing the spectral parameter θ_k as

$$\theta_k = \begin{cases} \theta_{k-1}^{N+}, & \text{if } \theta_k^{N+} \in [\frac{1}{4} + \eta, \tau], \\ 1, & \text{otherwise} \end{cases}$$

or

$$\theta_k = \begin{cases} \theta_{k-1}^{N-}, & \text{if } \theta_k^{N-} \in [\frac{1}{4}+\eta, \tau], \\ 1, & \text{otherwise,} \end{cases}$$

where η and τ are constants such that $\frac{1}{4}+\eta \le \theta_k \le \tau$,

$$\theta_k^{N-} = 1 - \frac{\|z_{k-1}\|^2 d_{k-1}^T g_k}{(d_{k-1}^T z_{k-1})(z_{k-1}^T g_k)}$$

and

$$\theta_k^{N+} = 1 - \frac{1}{z_{k-1}^T g_k}\left(\frac{\|z_{k-1}\|^2 d_{k-1}^T g_k}{d_{k-1}^T z_{k-1}} - s_{k-1}^T g_k\right).$$

Convergence of this method is established under the strong Wolfe conditions. For more conjugate gradient methods, the reader is referred to the works of [1, 2, 8, 15, 17, 18, 23–25, 27, 32, 33].

In this paper, we suggest another new hybrid conjugate gradient method that inherits good computational efforts of β_k^{LS} and β_k^{HS} methods and also nice convergence properties of β_k^{DY} and β_k^{CD} methods. This proposed method is presented in the next section, and the rest of the paper is structured as follows. In Sect. 3, we show that the proposed method satisfies the descent condition for any line search and also present its global convergence analysis under the strong Wolfe line search. Numerical comparison with respect to performance profiles of Dolan-Morè [7] and conclusion is presented in Sects. 4 and 5, respectively.

A new hybrid conjugate gradient method

In [32], a variant of the β_k^{PRP} method is proposed, where the coefficient β_k is computed as

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

This method inherits the good numerical performance of the PRP method. Moreover, Huang et al. [17] proved that the β_k^{WYL} method satisfies the sufficient descent property and established that the method is globally convergent under the strong Wolfe line search if the parameter σ in (5) satisfies $\sigma < \frac{1}{4}$. Yao et al. [34] extended this idea to the β_k^{HS} method and proposed the update

$$\beta_k^{YWH} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}.$$

The authors proved that the method is globally convergent under the strong Wolfe line search with the parameter $\sigma < \frac{1}{3}$. In Jian et al. [18], a hybrid of $\beta_k^{DY}, \beta_k^{FR}, \beta_k^{WYL}$ and β_k^{YWH} is proposed by introducing the update

$$\beta_k^N = \frac{\|g_k\|^2 - \max\{0, \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}\}}{\max\{\|g_{k-1}\|^2, d_{k-1}^T (g_k - g_{k-1})\}},$$

with

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

Independent of the line search, the method generates a descent direction at every iteration. Furthermore, its global convergence is established under the weak Wolfe line search.

Now, motivated by the ideas of Jian et al. [18], in this paper we suggest a hybrid conjugate gradient method that inherits the strengths of the β_k^{LS}, β_k^{HS}, β_k^{DY} and β_k^{CD} methods by introducing β_k^{PKT} as

$$\beta_k^{PKT} = \begin{cases} \dfrac{\|g_k\|^2 - g_k^T g_{k-1}}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}}, & \text{if } 0 < g_k^T g_{k-1} < \|g_k\|^2, \\ \dfrac{\|g_k\|^2}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}}, & \text{otherwise,} \end{cases} \tag{7}$$

with direction d_k defined as

$$d_k = \begin{cases} -g_k, & \text{if} \quad k = 0 \text{ or } |g_k^T g_{k-1}| \ge 0.2\|g_k\|^2, \\ -\left(1 + \beta_k^{PKT} \frac{d_{k-1}^T g_k}{\|g_k\|^2}\right) g_k + \beta_k^{PKT} d_{k-1}, & \text{if} \quad k > 0. \end{cases} \tag{8}$$

Now, with β_k and d_k defined as in (7) and (8), respectively, we present our hybrid conjugate gradient algorithm below.

Algorithm 1 A new hybrid conjugate gradient method

1: Give initial guess $x_0 \in \mathbb{R}^n$, parameters $0 < \delta < \sigma < 1$ and $\epsilon > 0$.
2: Set $k = 0$ and $d_0 = -g_0$. If $||g_0|| < \epsilon$ then stop.
3: **for** $k = 0, 1, \dots$ **do**
4: Compute α_k using the strong Wolfe conditions (5) and use the recurrence

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

to find the next iterate. Set $k = k + 1$.
5: Compute $g_k = \nabla f(x_k)$.
6: If $||g_k|| < \epsilon$ stop.
7: Compute β_k using (7).
8: Generate d_k by (8). Go to step 4.
9: **end for**

Global convergence of the proposed method

The following standard assumptions which have been used extensively in the literature are necessary to analyse the global convergence properties of our hybrid method.

Assumption 3.1 The level set

$$X = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\},$$

is bounded, where $x_0 \in \mathbb{R}^n$ is the initial guess of the iterative method (2).

Assumption 3.2 In some neighbourhood N of X, the objective function f is continuous and differentiable, and its gradient is Lipschitz continuous, that is, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \le L\|x - y\| \text{ for all } x, y \in N.$$

If Assumptions 3.1 and 3.2 hold, then there exists a positive constant ς such that

$$\|g(x)\| \le \varsigma \text{ for all } x \in N. \tag{9}$$

Lemma 3.1 *Consider the sequence $\{g_k\}$ and $\{d_k\}$ generated by Algorithm 1. Then, the sufficient descent condition*

$$d_k^T g_k = -\|g_k\|^2, \ \forall k \ge 0, \tag{10}$$

holds.

Proof If $k = 0$ or $|g_k^T g_{k-1}| \ge 0.2\|g_k\|^2$, then the search direction d_k is given by

$$d_k = -g_k.$$

This gives

$$g_k^T d_k = -\|g_k\|^2.$$

Otherwise, the search direction d_k is given by

$$d_k = -\left(1 + \beta_k^{PKT} \frac{d_{k-1}^T g_k}{\|g_k\|^2}\right) g_k + \beta_k^{PKT} d_{k-1}. \tag{11}$$

Now, if we pre-multiply Eq. (11) by g_k^T, we get

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

Therefore, the new method satisfies the sufficient descent property (10) for all k. □

Lemma 3.2 *Suppose that Assumptions 3.1 and 3.2 hold. Let the sequence $\{x_k\}$ be generated by (2) and the search direction d_k be a descent direction. If α_k is obtained by the strong Wolfe line search, then the Zoutendijk condition*

$$\sum_{k \ge 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \tag{12}$$

holds.

Lemma 3.3 *For any $k \ge 1$, the relation $0 < \beta_k^{PKT} \le \beta_k^{CD}$ always holds.*

Proof From (5) and (10), it follows that

$$d_{k-1}^T y_{k-1} \ge (1 - \sigma)\|g_{k-1}\|^2,$$

and since $0 < \sigma < 1$, we have

$$d_{k-1}^T y_{k-1} > 0. \tag{13}$$

Also, by descent condition (10), we get

$$-g_{k-1}^T d_{k-1} = \|g_{k-1}\|^2,$$

implying

$$-g_{k-1}^T d_{k-1} > 0. \tag{14}$$

Therefore, from (7), (13) and (14), it is clear that $\beta_k^{PKT} > 0$. Moreover, if $0 < g_k^T g_{k-1} < \|g_k\|^2$, then

$$\begin{aligned}\beta_k^{PKT} &= \frac{\|g_k\|^2 - g_k^T g_{k-1}}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}} \\ &< \frac{\|g_k\|^2}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}},\end{aligned}$$

and since $\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\} \geq -g_{k-1}^T d_{k-1}$, we have

$$\begin{aligned}\beta_k^{PKT} &\leq \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}} \\ &= \beta_k^{CD}.\end{aligned}$$

Hence, lemma is proved. □

Theorem 3.1 *Suppose that Assumptions 3.1 and 3.2 hold. Consider the sequences $\{g_k\}$ and $\{d_k\}$ generated by Algorithm 1. Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{15}$$

Proof Assume that (15) does not hold. Then there exists a constant $r > 0$ such that

$$\|g_k\| \geq r, \ \forall k \geq 0. \tag{16}$$

Letting $\xi_k = 1 + \beta_k^{PKT} \frac{d_{k-1}^T g_k}{\|g_k\|^2}$, it follows from (8) that

$$d_k + \xi_k g_k = \beta_k^{PKT} d_{k-1}.$$

Squaring both sides gives

$$\begin{aligned}&\|d_k\|^2 + 2\xi_k d_k^T g_k + \xi_k^2 \|g_k\|^2 = (\beta_k^{PKT})^2 \|d_{k-1}\|^2, \\ &\Rightarrow \|d_k\|^2 = (\beta_k^{PKT})^2 \|d_{k-1}\|^2 - 2\xi_k d_k^T g_k - \xi_k^2 \|g_k\|^2.\end{aligned}$$

Now, dividing by $(g_k^T d_k)^2$ and applying the descent condition $g_k^T d_k = -\|g_k\|^2$ yields

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} = (\beta_k^{PKT})^2 \frac{\|d_{k-1}\|^2}{(g_k^T d_k)^2} + \frac{2\xi_k}{\|g_k\|^2} - \frac{\xi_k^2}{\|g_k\|^2}.$$

Since $\beta_k^{PKT} \leq \beta_k^{CD} = \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}}$, we obtain

$$\begin{aligned}\frac{\|d_k\|^2}{(g_k^T d_k)^2} &\leq \left(\frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}}\right)^2 \frac{\|d_{k-1}\|^2}{(g_k^T d_k)^2} + \frac{2\xi_k}{\|g_k\|^2} - \frac{\xi_k^2}{\|g_k\|^2} \\ &= \frac{\|g_k\|^4}{(g_{k-1}^T d_{k-1})^2} \frac{\|d_{k-1}\|^2}{(g_k^T d_k)^2} + \frac{2\xi_k}{\|g_k\|^2} - \frac{\xi_k^2}{\|g_k\|^2} \\ &= \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{1}{\|g_k\|^2}(\xi_k^2 - 2\xi_k + 1 - 1) \\ &= \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{(\xi_k - 1)^2}{\|g_k\|^2} + \frac{1}{\|g_k\|^2} \\ &\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2}.\end{aligned} \tag{17}$$

Noting that

$$\frac{\|d_0\|^2}{(g_0^T d_0)^2} = \frac{1}{\|g_0\|^2},$$

and using (17) recursively yields

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \sum_{i=0}^{k} \frac{1}{\|g_i\|^2}. \tag{18}$$

From (16), we have

$$\sum_{i=0}^{k} \frac{1}{\|g_i\|^2} \leq \frac{k+1}{r^2}.$$

Thus,

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{r^2}{k+1},$$

which implies that

Table 1 Numerical results of the methods

Function	Dim	PKT				AZPRP				N			
		NI	FE	GE	CPU	NI	FE	GE	CPU	NI	FE	GE	CPU
Ext. QP2	10,000	6	399	171	0.307	37	289	191	0.243	**33**	**163**	**83**	**0.220**
Ext. Rosenbrock	10,000	**19**	**101**	**43**	**0.0169**	43	266	163	0.0762	71	317	149	0.0851
Ext. Penalty	500	**10**	**49**	**15**	**0.00339**	15	65	23	0.00615	24	103	25	0.00774
Ext. Beale	10,000	**14**	**43**	**26**	**0.134**	17	110	95	0.303	46	109	47	0.262
Ext. Wood	50,000	**43**	**218**	90	0.2	157	634	167	0.621	70	248	**71**	0.283
Ext. Denschnb	50,000	**8**	**24**	14	**0.0318**	9	20	**11**	0.0414	12	25	13	0.0484
Ext. Denschnf	20,000	13	71	31	0.0626	**12**	**49**	**16**	**0.0361**	24	86	25	0.0663
Ext. Himmelblau	50,000	**9**	**37**	**15**	**0.0409**	11	38	16	0.0495	13	41	**15**	0.0593
Ext. Powell	2000	**41**	**187**	**106**	**0.197**	224	693	345	0.836	947	2820	948	2.29
Ext TET	2000	6	17	11	0.00529	**5**	**14**	**9**	**0.00415**	9	19	10	0.00568
Perturbed Quad	500	**122**	611	241	0.116	**122**	**489**	**123**	**0.0688**	**122**	**489**	**123**	0.0695
DQDRTIC	10,000	10	41	14	0.0154	**5**	**19**	**7**	**0.00603**	32	109	33	0.0365
ARWHEAD	100	**6**	28	12	0.00204	7	**27**	**10**	**0.00192**	20	69	21	0.00437
QUARTC	7000	**3**	**22**	**20**	**0.0603**	10	81	80	0.178	82	449	448	1.14
Tridia	500	597	3477	1080	0.195	**225**	**1125**	**284**	**0.0614**	1496	7235	1497	0.396
LIARWHD	500	**14**	**82**	**28**	**0.00538**	26	112	32	0.00685	33	141	34	0.00836
ENGVAL1	500	**19**	54	31	0.00473	20	51	29	**0.00446**	21	**45**	**22**	0.00452
NONSCOMP	20,000	**38**	152	69	0.0782	38	135	64	**0.0680**	41	**127**	**47**	0.0758
Diagonal4	10,000	**6**	**13**	**7**	**0.00419**	6	14	8	0.0382	10	24	11	0.00871
Ext. Tridiagonal 2	1000	**26**	**54**	**28**	0.058	**26**	**54**	**28**	**0.00673**	29	56	30	0.0283
FLETCHCR	1000	**2012**	**13350**	**4336**	1.13	2986	13688	4702	1.17	4276	17273	4339	1.36
NONDIA	20,000	**7**	**55**	**8**	0.11	32	238	60	**0.105**	141	998	144	0.34
CUBE	500	**1640**	**8511**	**2831**	**1.86**	5898	26124	8561	6.05	-	-	-	-
Ext. Tridiagonal 1	20,000	**12**	**63**	**53**	**0.302**	17	76	70	0.386	334	464	450	2.95
SINQUAD	800	120	740	**327**	0.158	**118**	**737**	429	**0.0988**	379	1577	401	0.182
Almost Perturbed Quad	20,000	**804**	**4829**	**808**	**1.7**	1339	6983	1625	2.12	1314	6677	1315	3.3
Perturbed Trid Quad	50,000	1201	9608	2435	9.81	**1201**	**7207**	**1202**	**6.76**	**1201**	**7207**	**1202**	**7.53**
Cosine	5000	**8**	**27**	**19**	**0.0196**	**8**	**27**	**19**	0.0271	19	41	22	0.0533
FHess1	50	**246**	**934**	**514**	**0.0686**	404	1160	546	0.0894	1368	4028	1372	0.303
FHess2	500	**882**	**5536**	**1202**	**0.775**	4276	25496	4530	3.11	4166	24,977	4167	3.13
FHess3	10,000	**2**	**13**	**3**	0.00619	**2**	**13**	**3**	0.00588	**2**	**13**	**3**	**0.00525**
Ext. BD1	50,000	**9**	**35**	**22**	**0.101**	11	41	29	0.103	17	46	28	0.122
Perturbed Quad Diagonal	100,000	**343**	**1356**	**393**	**39.9**	2334	11467	2452	248	1907	9528	1908	193
Gen. Quartic	50,000	9	26	16	0.0379	**6**	**14**	**8**	**0.0299**	7	15	**8**	0.031
Quadratic QF1	500	**122**	610	362	0.0411	**122**	**489**	241	**0.0317**	196	676	**197**	0.0468
Quadratic QF2	500	**214**	1083	224	0.0615	221	893	230	**0.0469**	218	**873**	**219**	0.132
Diagonal 5	5000	**2**	**5**	**5**	0.0105	4	**5**	**5**	0.00479	4	**5**	**5**	**0.00447**
Diagonal 5	1000	**2**	5	5	0.00339	3	**4**	**4**	**0.0013**	3	**4**	**4**	0.00152
Diagonal 2	5000	**280**	**1341**	**1332**	**0.965**	430	7187	7186	3.31	5623	12829	12,829	10.7
Gen. Tridiagonal 1	1000	**19**	**57**	**24**	0.00676	20	60	**24**	**0.00663**	26	73	27	0.0088
Gen. Tridiagonal 2	1000	39	138	52	0.0108	**32**	**105**	41	**0.00812**	35	106	**36**	0.00888
Gen. PSC1	1000	**12**	**93**	**85**	**0.0108**	17	**93**	90	0.0123	76	387	383	0.0507
Ext. PSC1	1000	**7**	26	15	0.00389	9	**21**	**11**	**0.00303**	10	22	**11**	0.00336
Dixon3dq	5000	**2500**	**5012**	**2513**	**1.10**	5000	10010	5011	1.95	5000	10007	5008	2.47
Ext. Quad Penalty QP1	500	**6**	**27**	**16**	**0.00228**	9	45	29	0.00395	15	36	17	0.00337
Biggsb1	500	**500**	1007	508	0.100	**500**	1007	508	**0.0926**	**500**	**1004**	**505**	0.0988
Ext. White & Holst	400	**35**	**202**	**86**	0.0266	43	203	96	**0.0265**	148	573	149	0.0665
NONDQUAR	1000	**1006**	**3739**	**3463**	**2.82**	1691	4045	3497	3.49	5747	7715	7201	7.44

Table 1 (continued)

Function	Dim	PKT				AZPRP				N			
		NI	FE	GE	CPU	NI	FE	GE	CPU	NI	FE	GE	CPU
Raydan2	200	**3**	**5**	**5**	0.00113	**3**	**5**	**5**	**0.000969**	56	6	6	0.00112
BDQRTIC	50	**59**	**250**	**85**	**0.0248**	102	402	132	0.055	106	386	107	0.0423
Raydan1	200	**84**	180	94	0.0220	85	175	88	0.0146	**84**	**171**	**85**	**0.0146**
Gen. White & Holst	50	1243	7061	1990	0.625	**1174**	**5189**	1679	**0.373**	1461	5830	**1467**	0.424
Ext Quad Exponential EP1	50	**2**	**11**	**3**	0.00111	3	13	4	**0.000937**	3	13	4	0.00110
Diagonal1	10	**18**	48	27	0.0433	**18**	63	45	0.027	19	**39**	**20**	**0.0204**
Diagonal2	5000	**280**	**1323**	**1318**	**1.01**	471	2102	2101	1.48	5703	13,034	13,034	12.5

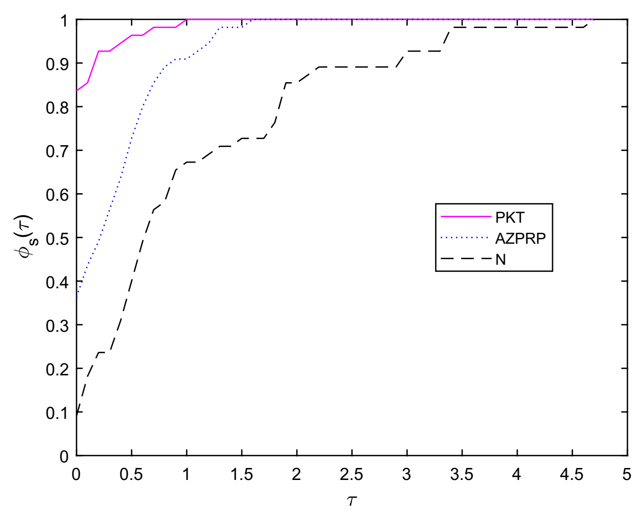

Fig. 1 Iterations performance profile

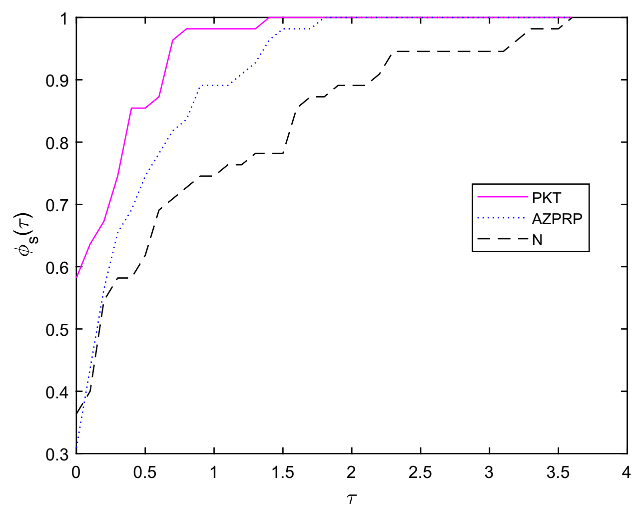

Fig. 2 Gradient evaluations performance profile

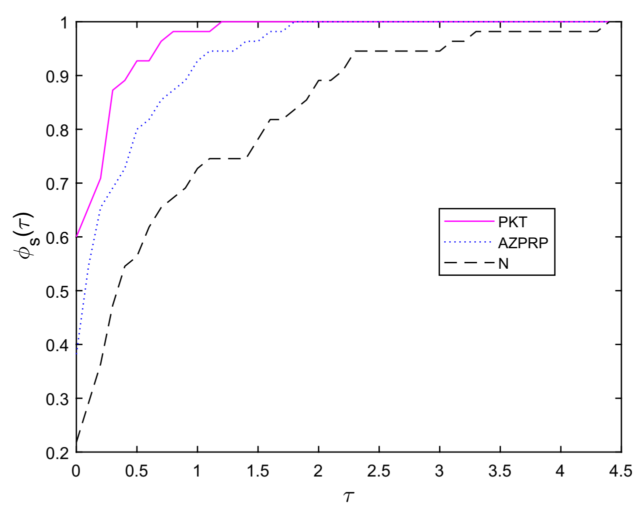

Fig. 3 Function evaluations performance profile

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq r^2 \sum_{k=0}^{\infty} \frac{1}{k+1} = +\infty.$$

This contradicts the Zoutendjik condition (12), concluding the proof. □

Numerical results

In this section, we analyse the numerical efficiency of our proposed β_k^{PKT} method, herein denoted PKT, by comparing its performance to that of Jian et al. [18], herein denoted N, and that of Alhawarat et al. [3], herein denoted AZPRP, on a set of 55 unconstrained test problems selected from [4]. We stop the iterations if either $\|g_k\| \leq 10^{-5}$ or a maximum of 10,000 iterations is reached. All the algorithms are coded

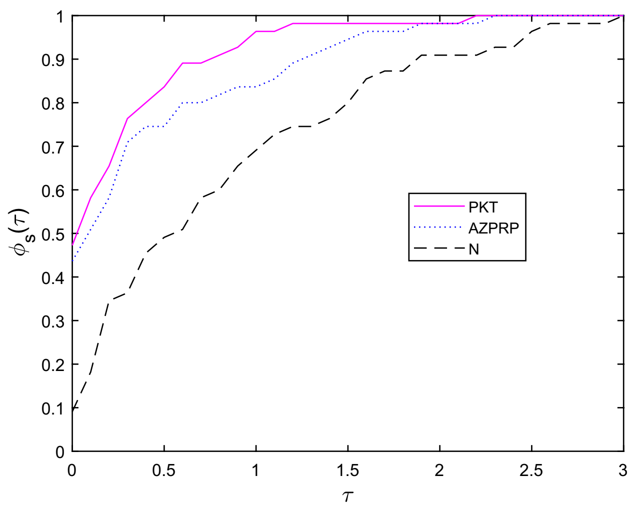

Fig. 4 Cpu performance profile

in MATLAB R2019a. The authors for both N and AZPRP methods suggested that their algorithms will have optimal performance if they are implemented with the generalized Wolfe line search conditions (6) with the following choice of parameters: $\sigma = 0.1$, $\delta = 0.0001$ and $\sigma_1 = 1 - 2\delta$ for N method and $\sigma = 0.4$, $\delta = 0.0001$ and $\sigma_1 = 0.1$ for AZPRP method. Hence for our numerical experiments, we set the parameters of N and AZPRP as determined in the respective papers. For the PKT method, we implemented the strong Wolfe line search conditions with $\delta = 0.0001$ and $\sigma = 0.05$.

Numerical results are presented in Table 1, where "Function" denotes name of test problem, "Dim" denotes dimension of test problem,"NI" denotes number of iterations, "FE" denotes number of function evaluations, "GE" denotes number of gradient evaluations, "CPU" denotes CPU time in seconds and "-" means that the method failed to solve the problem within 10,000 iterations. The bolded figures show the best performer for each problem. From Table 1, we observe that the PKT and AZPRP methods successfully solved all the problems, whereas the N method failed to solve one problem within 10,000 iterations. Moreover, the numerical results in the table indicate that the new PKT method is competitive as it is the best performer for a significant number of problems.

To further illustrate the performance of the three methods, we adopted the performance profile tool proposed by Dolan and Morè [7]. This tool evaluates and compares the performance of n_s solvers running on a set of n_p problems. The comparison between the solvers is based on the performance ratio

$$r_{p,s} = \frac{f_{p,s}}{\min(f_{p,i} : 1 \leq i \leq n_s)}, \tag{19}$$

where $f_{p,s}$ denotes either number of functions (gradient) evaluations, number of iterations or CPU time required by solver s to solve problem p. The overall evaluation of the performance of the solvers is then given by the performance profile function

$$\phi_s(\tau) = \frac{1}{n_p}\text{size}\{p : 1 \leq p \leq n_p,\ \ln(r_{p,s}) \leq \tau\}, \tag{20}$$

where $\tau \geq 0$. If solver s fails to solve a problem p, we set the ratio $r_{p,s}$ to some sufficiently large number.

The corresponding profiles are plotted in Figs. 1, 2, 3 and 4, where Fig. 1 shows the performance profile of number of iterations, Fig. 2 shows the performance profile of number of gradient evaluations, Fig. 3 shows the performance profile of function evaluations and Fig. 4 shows the performance profile of CPU time. The figures illustrate that the new method outperforms the AZPRP and N conjugate gradient methods.

Conclusion

In this paper, we developed a new hybrid conjugate gradient method that inherits the features of the famous Liu and Storey (LS), Hestenes and Stiefel (HS), Dai and Yuan (DY) and Conjugate Descent (CD) conjugate gradient methods. The global convergence of the proposed method was established under the strong Wolfe line search conditions. We compared the performance of our method with those of Jian et al. [18] and Alhawarat et al. [3] on a number of benchmark unconstrained optimization problems. Evaluation of performance based on the tool of Dolan-Moré [7] showed that the proposed method is both efficient and effective.

References

1. Abbo, K.K., Hameed, N.H.: New hybrid conjugate gradient method as a convex combination of Liu-Storey and Dixon methods. J. Multidiscip. Model. Optim. **1**(2), 91–99 (2018)
2. Abdullahi, I., Ahmad, R.: Global convergence analysis of a new hybrid conjugate gradient method for unconstrained optimization problems. Malays. J. Fundam. Appl. Sci. **13**(2), 40–48 (2017)
3. Alhawarat, A., Salleh, Z., Mamat, M., Rivaie, M.: An efficient modified Polak-Ribière-Polyak conjugate gradient method with global convergence properties. Optim. Methods Softw. **32**(6), 1299–1312 (2017)

4. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
5. Babaie-Kafaki, S.: A quadratic hybridization of Polak-Ribière-Polyak and Fletcher-Reeves conjugate gradient methods. J. Optim. Theory Appl. **154**(3), 916–932 (2012)
6. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)
7. Dolan, E.D., Morè, J.J.: Benchmarking optimization software with performance profiles, Math. Program., **91**, 201–214 (2002)
8. Djordjević, S.S.: New hybrid conjugate gradient method as a convex combination of LS and FR methods. Acta Math. Sci. **39**(1), 214–228 (2019)
9. Faramarzi, P., Amini, K.: A modified spectral conjugate gradient method with global convergence. J. Optim. Theory Appl. **182**(2), 667–690 (2019)
10. Fletcher, R.: Practical Methods of Optimization, Unconstrained Optimization. Wiley, New York (1987)
11. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**(2), 149–154 (1964)
12. Gu, B., Sun, X., Sheng, V.S.: Structural minimax probability machine. IEEE Trans. Neural Netw. Learn. Syst. **28**(7), 1646–1656 (2017)
13. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**(1), 170–192 (2005)
14. Hager, W.W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. **23**(4), 2150–2168 (2013)
15. Hassan, B.A., Alashoor, H.A.: A new hybrid conjugate gradient method with guaranteed descent for unconstraint optimization. Al-Mustansiriyah J. Sci. **28**(3), 193–199 (2017)
16. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bureau Stand. **49**(6), 409–436 (1952)
17. Huang, H., Wei, Z., Yao, S.: The proof of the sufficient descent condition of the Wei–Yao–Liu conjugate gradient method under the strong Wolfe–Powell line search. Appl. Math. Comput. **189**(2), 1241–1245 (2007)
18. Jian, J., Han, L., Jiang, X.: A hybrid conjugate gradient method with descent property for unconstrained optimization. Appl. Math. Model. **39**, 1281–1290 (2015)
19. Khakrah, E., Razani, A., Mirzaei, R., Oveisiha, M.: Some metric characterization of well-posedness for hemivariational-like inequalities. J. Nonlinear Funct. Anal. **2017**, 44 (2017)
20. Khakrah, E., Razani, A., Oveisiha, M.: Pascoletti–Serafini scalarization and vector optimization with a vaiable ordering structure. J. Stat. Manag. Syst. **21**(6), 917–931 (2018)
21. Khakrah, E., Razani, A., Oveisiha, M.: Pseudoconvex multiobjective continuous-time problems and vector variational inequalities, Int. J. Ind. Math. **9**(3) (2017), Article ID IJIM-00861, 8 pages
22. 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)
23. Li, X., Zhao, X.: A hybrid conjugate gradient method for optimization problems. Nat. Sci. **3**, 85–90 (2011)
24. Liu, J.K., Li, S.J.: New hybrid conjugate gradient method for unconstrained optimization. Appl. Math. Comput. **245**, 36–43 (2014)
25. Li, W., Yang, Y.: A nonmonotone hybrid conjugate gradient method for unconstrained optimization. J. Inequal. Appl. **2015**, 124 (2015). https://doi.org/10.1186/s13660-015-0644-1
26. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, Part 1: theory. J. Optim. Theory Appl. **69**(1), 129–137 (1991)
27. Mtagulwa, P., Kaelo, P.: An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems. Appl. Numer. Math. **145**, 111–120 (2019)
28. Polak, E., Ribiére, G.: Note sur la convergence de directions conjugées, Rev. Francaise Informat Recherche Opèrationelle, 3e Année, **16**, 35–43 (1969)
29. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**(4), 94–112 (1969)
30. Salih, Y., Hamoda, M.A., Rivaie, M.: New hybrid conjugate gradient method with global convergence properties for unconstrained optimization. Malays. J. Comput. Appl. Math. **1**(1), 29–38 (2018)
31. Touati-Ahmed, D., Storey, C.: Efficient hybrid conjugate gradient techniques. J. Optim. Theory Appl. **64**(2), 379–397 (1990)
32. Wei, Z., Yao, S., Liu, L.: The convergence properties of some new conjugate gradient methods. Appl. Math. Comput. **183**(2), 1341–1350 (2006)
33. Xu, X., Kong, F.: New hybrid conjugate gradient methods with the generalized Wolfe line search. SpringerPlus **5**(1), 881 (2016). https://doi.org/10.1186/s40064-016-2522-9
34. Yao, S., Wei, Z., Huang, H.: A note about WYL's conjugate gradient method and its application. Appl. Math. Comput. **191**(2), 381–388 (2007)
35. Yuan, G., Meng, Z., 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)
36. Zhou, Z., Wang, Y., Wu, Q.M.J., Yang, C.N., Sun, X.: Effective and efficient global context verification for image copy detection. IEEE Trans. Inf. Forensics Secur. **12**(1), 48–63 (2017)