Journal of Applied Mathematics and Computing (2023) 69:2531–2548
https://doi.org/10.1007/s12190-022-01821-z

ORIGINAL RESEARCH

A new hybrid conjugate gradient algorithm based on the Newton direction to solve unconstrained optimization problems

Naima Hamel[1] · **Noureddine Benrabia**[1,2] · **Mourad Ghiat**[1] · **Hamza Guebbai**[1]

Received: 15 July 2022 / Revised: 12 November 2022 / Accepted: 20 November 2022 /
Published online: 27 February 2023

Abstract

In this paper, we propose a new hybrid conjugate gradient method to solve unconstrained optimization problems. This new method is defined as a convex combination of DY and DL conjugate gradient methods. The special feature is that our search direction respects Newton's direction, but without the need to store or calculate the second derivative (the Hessian matrix), due to the use of the secant equation that allows us to remove the troublesome part required by the Newton method. Our search direction not only satisfies the descent property, but also the sufficient descent condition through the use of the strong Wolfe line search, the global convergence is proved. The numerical comparison shows the efficiency of the new algorithm, as it outperforms both the DY and DL algorithms.

Keywords Unconstraind optimization · Conjugate gradient method · Newton direction · Hybrid method · Global convergence

Noureddine Benrabia, Mourad Ghiat and Hamza Guebbai contributed equally to this work.

✉ Mourad Ghiat
ghiat.mourad@univ-guelma.dz; mourad.ghi24@gmail.com

Naima Hamel
hamel.naima@univ-guelma.dz; hamelnaima24@gmail.com

Noureddine Benrabia
n.benrabia@univ-soukahras.dz; noureddinebenrabia@yahoo.fr

Hamza Guebbai
guebbai.hamza@univ-guelma.dz; guebaihamza@yahoo.fr

[1] Laboratoire de Mathématiques Appliquées et de Modélisation, Université 8 Mai 1945 Guelma, B.P. 401, 24000 Guelma, Algeria

[2] Département de Mathématiques et Informatique, Université Mohamed-Chérif Messaadia, B.P. 1553, 41000 Souk Ahras, Algeria

Mathematics Subject Classification 90C06 · 90C26 · 90C30 · 65K05 · 49M15

1 Introduction

In this study, we consider the unconstrained optimization problems formulated as follows:

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

where, $f:\mathbb{R}^n \mapsto \mathbb{R}$ is a continuously differentiable function and its gradient is denoted by $g(x) = \nabla f(x)$. The numerical techniques for solving (1) are iterative, specifically, starting with an appropriate initial vector $x_0 \in \mathbb{R}^n$, the iterations are generated by this reccurence relation :

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k \geq 0, \tag{2}$$

where, α_k is the step size determined using an exact/inexacte line search technique, and d_k is the search direction supposed to satisfy the descent property

$$g_k^T d_k < 0, \quad k \geq 0,$$

or the suffcient descent condition

$$g_k^T d_k \leq -C \parallel g_k \parallel^2, \quad k \geq 0,$$

where $C > 0$ [1]. The step size α_k is mostly chosen to satisfy the following famous strong Wolfe inexact line search [2, 3]

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k \tag{3}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{4}$$

where, $0 < \delta < \sigma < \frac{1}{2}$.

Depending on the calculation of the search direction, there are several methods to solve (1), in this study we are interested by the Newton method and the conjugate gradient methods. The search direction of the Newton method is calculated as follows

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{5}$$

where, $\nabla^2 f(x_{k+1})$ is the Hessian matrix of f. The Newton method uses the second derivative information (Hessian matrix) to update the search direction d_k which allows it to give a quadratic convergence rate, but in practice, especially when n is large, methods that do not require Hessian evaluation are preferred over those needing it [4]. The conjugate gradient method does not require much storage space compared to other methods, because it only needs the first derivative information, hence it is very practical for solving large-scale unconstrained optimization problems [4], the search direction given as:

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k. \tag{6}$$

Depending on the choice of the parameter $\beta_k \in \mathbb{R}$ known as the conjugate gradient parameter, there are several different conjugate gradient algorithms. In the following, we recall some famous formulas for this parameter:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \text{(HS-Hestenes and Stiefel [5])},$$

$$\beta_k^{FR} = \frac{\| g_{k+1} \|^2}{\| g_k \|^2}, \quad \text{(FR-Fletcher and Reeves [6])},$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\| g_k \|^2}, \quad \text{(PRP-Polak and Ribire [7, 8])},$$

$$\beta_k^{CD} = \frac{\| g_{k+1} \|^2}{-d_k^T g_k}, \quad \text{(CD-conjugate descent [9])},$$

$$\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \quad \text{(LS-Liu and Storey [10])},$$

$$\beta_k^{DY} = \frac{\| g_{k+1} \|^2}{d_k^T y_k}, \quad \text{(DY-Dai and Yuan[11])},$$

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k}, \quad \text{(DL - Dai and Liao [12])}.$$

Where, $\| \cdot \|$ is the Euclidean norm, $t \geq 0$, $y_k = g_{k+1} - g_k$ and $s_k = \alpha_k d_k$. In the linear case, i.e. if the objective function is quadratic and α_k satisfies the exact line search, DY, FR, CD, PRP, HS, LS conjugate gradient methods are identical. Therefore, the convergence results are similar. For general nonlinear functions, the convergence results are related to the selection of the parameter β_k and the type of the line search applied. DL conjugate gradient method is considired as a modification of the Hestenes and Stiefel (HS) method [12] and one of the best performing conjugate gradient methods.

Many researchers have tried to devise new methods based on hybrid technics, which are considered more efficient than the original methods, because they aim to integrate the strengths and good performance of the methods to be combined. So, several hybrid methods are suggested, for example, in [13] Yao and Qin suggested a nonlinear conjugate gradient method which can be viewed as a hybrid of DL and WYL conjugate gradient methods. Xu and Kong [14] also suggested two hybrid methods, the first one is a linear combination between DY and HS conjugate gradient methods and the second one is between FR and PRP. Following the work done by Xu and Kong [14], Narayanan and Kaelo [4] developed a new hybrid conjugate gradient method as a linear hybridization between the DY method and the HS method or its modifications.

Djordjevic [15] proposed a hybrid conjugate gradient method by using the convex combination of FR and PRP methods, the search direction satisfies the conjugacy condition, see also [16–21]. Andrai [22] introduced another hybrid conjugate gradient method with β_k computed as a convex combination of DY and HS methods, the special feature of this hybrid method is that the search direction is the Newton direction and it outperforms many other conjugate gradient methods. This idea has inspired many researchers to devise new hybrid methods, see [1, 23–29]. Note that, DY method has strong convergence results but poor behavior and in general, the HS method may not converge, but in practice it is very efficient. As we know, DL conjugate gradient method is the modification of the HS method [12] and one of the best performing conjugate gradient methods. For these reasons and motivated by Andrai's idea [22], we suggest to combine DY and DL methods as a new hybrid conjugate gradient algorithm to solve (1) by computing the conjugate gradient parameter β_k denoted in this paper by β_k^{HBGG} as a convex combination of DY and DL formulas:

$$\beta_k^{HBGG} = (1-\theta_k)\beta_k^{DL} + \theta_k \beta_k^{DY},$$

where, $\theta_k \in [0, 1]$, and we seek to combine some good properties of both our conjugate gradient method and Newton's method by making our search direction represents Newton's search direction, without computing or storing the second derivative required by Newton's method.

The paper is organized in the following manner. In Sect. 2, we build the new method and obtain the value of the parameter θ_k, also we present the algorithm of our method and prove that, under a strong Wolfe line search , the descent property and the sufficient descent condition hold. In Sect. 3, the global convergence of the new method is proved. Then we discuss the numerical results in Sect. 4. Finally the conclusions are provided in Sect. 5.

2 The new hybrid conjugate gradient method

In this section, we introduce our hybrid conjugate gradient method as a convex combination of DY and DL algorithms by defining the search direction as follows:

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k^{HBGG} d_k, \tag{7}$$

where

$$\beta_k^{HBGG} = (1-\theta_k)\beta_k^{DL} + \theta_k \beta_k^{DY}. \tag{8}$$

So, we can write

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + (1-\theta_k)\beta_k^{DL} d_k + \theta_k \beta_k^{DY} d_k, \tag{9}$$

where $\theta_k \in [0, 1]$. If $\theta_k = 0$, then $\beta_k^{HBGG} = \beta_k^{DL}$ and if $\theta_k = 1$, then $\beta_k^{HBGG} = \beta_k^{DY}$. On the other hand if $0 < \theta_k < 1$ then β_k^{HBGG} is the convex combination between β_k^{DY} and β_k^{DL}.

Assume that $\nabla^2 f(x)^{-1}$ exists at each iterative point for the objective function f. As we know, the Newton method has quadratic convergence property for solving (1), this partly depends on its search direction, so we are going to build a new hybrid conjugate gradient method having some good properties of the Newton method. To achieve this, we compute the scalar θ_k such that our search direction given by (9) is equal to the Newton direction, i.e.

$$-g_{k+1} + (1-\theta_k)\beta_k^{DL} d_k + \theta_k \beta_k^{DY} d_k = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}. \tag{10}$$

This idea is similar to that of Andrai, see [22].
Multiplying both sides of the equation (10) by $s_k^T \nabla^2 f(x_{k+1})$ from the left we obtain:

$$-s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1-\theta_k)\beta_k^{DL} s_k^T \nabla^2 f(x_{k+1}) d_k + \theta_k \beta_k^{DY} s_k^T \nabla^2 f(x_{k+1}) d_k \\ = -s_k^T g_{k+1}. \tag{11}$$

Assume that the pair (s_k, y_k) satisfies the following secant condition

$$\nabla^2 f(x_{k+1}) s_k = y_k,$$

i.e.

$$s_k^T \nabla^2 f(x_{k+1}) = y_k^T.$$

Then (11) becomes

$$-y_k^T g_{k+1} + (1-\theta_k)\beta_k^{DL} y_k^T d_k + \theta_k \beta_k^{DY} y_k^T d_k = -s_k^T g_{k+1}.$$

After some algebraic calculations, we get

$$\begin{aligned}
\theta_k &= \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - \beta_k^{DL} y_k^T d_k}{(-\beta_k^{DL} + \beta_k^{DY}) y_k^T d_k} \\
&= \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - g_{k+1}^T y_k + t g_{k+1}^T s_k}{(-g_{k+1}^T (g_{k+1} - g_k) + t g_{k+1}^T s_k + \| g_{k+1} \|^2} \\
&= \frac{-s_k^T g_{k+1} + t g_{k+1}^T s_k}{-g_{k+1}^T g_{k+1} + g_{k+1}^T g_k + t g_{k+1}^T s_k + \| g_{k+1} \|^2} \\
&= \frac{s_k^T g_{k+1}(t-1)}{g_{k+1}^T (g_k + t s_k)}.
\end{aligned} \tag{12}$$

Clearly, although we have calculated the scalar θ_k so that the direction (9) is the Newton direction, our algorithm does not require to calculate or store the second derivative (the Hessian matrix) required by the classical Newton method, and this is due to the use of the secant equation. Observe that, the θ_k parameter in (12) may be not in the interval [0, 1] for some iterations. Therefore, to have a real convex combination in (8), we consider this rule: If $\theta_k \leq 0$ then take $\theta_k = 0$ in (8), i.e. $\beta_k^{HBGG} = \beta_k^{DL}$, if $\theta_k \geq 1$, then take $\theta_k = 1$ in (8), i.e. $\beta_k^{HBGG} = \beta_k^{DY}$.

Now, we present our HBGG algorithm which has some nice features of both conjugate gradient algorithm and Newton's algorithm.

Algorithm 1 HBGG algorithm

1: Choose the initial point $x_0 \in \mathbb{R}^n$, $\epsilon > 0$.
2: Choose the parameter $t > 1$.
3: Calculate $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$.
4: Set $d_0 = -g_0$, the initial guess $\alpha_0 = \frac{1}{\|g_0\|}$. Let $k = 0$.
5: Test a criterion to stop the iterations
6: **if** $\| g_k \| \leq \epsilon$ **then**
7: Stop
8: **else**
9: go to step 11
10: **end if**
11: Compute the step size α_k by the strong Wolfe conditions (3), (4).
12: Updating the next iterate by: $x_{k+1} = x_k + \alpha_k d_k$.
13: Compute $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.
14: **if** $g_{k+1}^T(g_k + t s_k) = 0$ **then**
15: $\theta_k = 0$
16: **else**
17: calculate θ_k as in (12)
18: **end if**
19: **if** $\theta_k \leq 0$ **then**
20: $\beta_k^{HBGG} = \beta_k^{DL}$.
21: **else if** $\theta_k \geq 1$**then**
22: $\beta_k^{HBGG} = \beta_k^{DY}$.
23: **else**
24: calculate β_k^{HBGG} as in (8).
25: **end if**
26: Compute $d_{k+1} = -g_{k+1} + \beta_k^{HBGG} d_k$.
27: Set the initial guess $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$ and let $k = k + 1$.
28: go to step 6

Remark 1 If $\beta_k^{HBGG} = \beta_k^{DL}$ or $\beta_k^{HBGG} = \beta_k^{DY}$, then in these two cases, please refer to [12] and [11] respectively.

Theorem 1 *Let θ_k be given by (12) and supposed $0 < \theta_k < 1$, assume that $t > 1$ and α_k in Algorithm 1 is determined by the strong Wolfe line search (3), (4), then the direction defined by (9) is a descent direction, i.e. $g_k^T d_k < 0$.*

Proof The proof is done by recurrence. For k=0 : $g_0^T d_0 = -g_0^T g_0 = -\| g_0 \|^2 < 0$. Suppose that $g_k^T d_k < 0$ is satisfied for k $\geq$ 1 and show that it is satisfied for k + 1. Multiply (9) by g_{k+1} we find

$$
\begin{aligned}
g_{k+1}^T d_{k+1} &= - \parallel g_{k+1} \parallel^2 + (1-\theta_k)\beta_k^{DL} g_{k+1}^T d_k + \theta_k \beta_k^{DY} g_{k+1}^T d_k, \\
&= - \parallel g_{k+1} \parallel^2 + (1-\theta_k)\frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k} g_{k+1}^T d_k + \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \\
&= - \parallel g_{k+1} \parallel^2 + (1-\theta_k)\frac{g_{k+1}^T ((g_{k+1} - g_k) - t s_k)}{d_k^T y_k} g_{k+1}^T d_k \\
&\quad + \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \\
&= - \parallel g_{k+1} \parallel^2 + \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k \\
&\quad - \theta_k \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \\
&= - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2 - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k - \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k \\
&\quad - \theta_k \frac{- g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k + \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \\
&= - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2 - g_{k+1}^T g_k - t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k \\
&\quad + \theta_k \frac{g_{k+1}^T g_k + t g_{k+1}^T s_k}{d_k^T y_k} g_{k+1}^T d_k.
\end{aligned}
$$

We substitute θ_k in the above relation by (12) to obtain

$$
\begin{aligned}
g_{k+1}^T d_{k+1} &= - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{g_{k+1}^T (g_k + t s_k)}{d_k^T y_k} g_{k+1}^T d_k \\
&\quad + (t-1)\frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \qquad (13)
\end{aligned}
$$

From (12) we have

$$
\frac{g_{k+1}^T (g_k + t s_k)}{s_k^T g_{k+1} (t-1)} = \frac{1}{\theta_k},
$$

then

$$
g_{k+1}^T (g_k + t s_k) = \frac{(t-1) s_k^T g_{k+1}}{\theta_k}.
$$

Therefore (13) becomes

$$g_{k+1}^T d_{k+1} = - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{\theta_k d_k^T y_k} g_{k+1}^T d_k + (t-1) \frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \quad (14)$$

Observe that, since $s_k = \alpha_k d_k$ then

$$\frac{(t-1)\alpha_k (d_k^T g_{k+1})^2}{d_k^T y_k} = \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k. \quad (15)$$

Using strong Wolfe line search (4) we get

$$d_k^T y_k = g(x_k + \alpha_k d_k)^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k > 0. \quad (16)$$

Taking $t > 1$, since $\alpha_k > 0$, $d_k^T y_k > 0$ and from (15) we conclude that

$$\frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k > 0. \quad (17)$$

Knowing that $\frac{1}{\theta_k} > 1$, and from (17) it results

$$\frac{(t-1) s_k^T g_{k+1}}{\theta_k d_k^T y_k} g_{k+1}^T d_k > \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k,$$

$$-\frac{(t-1) s_k^T g_{k+1}}{\theta_k d_k^T y_k} g_{k+1}^T d_k < -\frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k,$$

$$- \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{\theta_k d_k^T y_k} g_{k+1}^T d_k + (t-1) \frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k$$
$$< - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k + (t-1) \frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k.$$

Therefore (14) becomes

$$g_{k+1}^T d_{k+1} < - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k - \frac{(t-1) s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k + (t-1) \frac{s_k^T g_{k+1}}{d_k^T y_k} g_{k+1}^T d_k,$$
$$= - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \quad (18)$$

$$= -(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) \parallel g_{k+1} \parallel^2 . \tag{19}$$

Since $d_k^T y_k > 0$ and $d_k^T g_k < 0$, then $g_{k+1}^T d_k = d_k^T y_k + d_k^T g_k < d_k^T y_k$, and

$$g_{k+1}^T d_k < d_k^T y_k \Longrightarrow \frac{g_{k+1}^T d_k}{d_k^T y_k} < 1,$$
$$-(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) < 0.$$

Then (19) becomes

$$g_{k+1}^T d_{k+1} < -(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}) \parallel g_{k+1} \parallel^2 < 0. \tag{20}$$

Therefore $g_k^T d_k < 0$ which shows that d_k is a descent direction. □

Theorem 2 *Let θ_k be given by (12) and supposed $0 < \theta_k < 1$, assume that $t > 1$ and α_k in Algorithm 1 is determined by the strong Wolfe line search (3), (4), then there exists a constante $C > 0$ such that the sufficient descent condition*

$$g_{k+1}^T d_{k+1} \le -C \parallel g_{k+1} \parallel^2 \tag{21}$$

holds.

Proof Multiply (9) by g_{k+1}, we get

$$\begin{aligned} g_{k+1}^T d_{k+1} &= - \parallel g_{k+1} \parallel^2 + (1-\theta_k)\beta_k^{DL} g_{k+1}^T d_k + \theta_k \beta_k^{DY} g_{k+1}^T d_k, \\ &= - \parallel g_{k+1} \parallel^2 + (1-\theta_k)\frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k} g_{k+1}^T d_k + \theta_k \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k. \end{aligned}$$

In the same way as in the previous proof, we obtain the relation (18):

$$\begin{aligned} g_{k+1}^T d_{k+1} &< - \parallel g_{k+1} \parallel^2 + \frac{\parallel g_{k+1} \parallel^2}{d_k^T y_k} g_{k+1}^T d_k, \\ &\le - \parallel g_{k+1} \parallel^2 + \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \parallel g_{k+1} \parallel^2 . \end{aligned} \tag{22}$$

From the second strong Wolfe condition (4), it holds

$$|g_{k+1}^T d_k| \le -\sigma g_k^T d_k \tag{23}$$

i.e.

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

Then

$$d_k^T y_k = g_{k+1}^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k > 0$$

$$\Longrightarrow \frac{1}{d_k^T y_k} \leq \frac{1}{-(1-\sigma) g_k^T d_k}. \tag{24}$$

From (23) and (24) it results

$$\frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \leq \frac{\sigma}{1-\sigma}.$$

Then (22) becomes

$$\begin{aligned} g_{k+1}^T d_{k+1} &\leq - \parallel g_{k+1} \parallel^2 + \frac{\sigma}{1-\sigma} \parallel g_{k+1} \parallel^2 \\ &\leq -(1 - \frac{\sigma}{1-\sigma}) \parallel g_{k+1} \parallel^2 . \end{aligned} \tag{25}$$

It suffices to take $C = \left(1 - \frac{\sigma}{1-\sigma}\right) > 0$, because $0 < \sigma < \frac{1}{2}$. Hence, the sufficient descent condition holds. □

In the Dai and Liao method, the conjugate gradient parameter is given by $\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k}$, where t is a positive parameter ($t \geq 0$). In our article, we have combined the formulas β_k^{DL} and β_k^{DY} as a convex combination, as shown in (8) to build a new hybrid conjugate gradient algorithm. Clearly, the parameter t appears in (8) and (12), so we consider t as a positive parameter ($t \geq 0$). But we found that, we should only take values of $t > 1$ to ensure the descent condition of our direction at each iteration, as shown in Theorem 1 and Theorem 2. Then, in Algorithm 1, we are restricted to take values of $t > 1$. Since $t > 1$ so we have a large choice for the values of t, we will choose the optimal value that allows Algorithm 1 to perform well by making a comparison between some values of t. A detailed discussion will be described later in Sect. 4.

3 Convergence analysis

The following essential assumptions on the objective function are required to establish the global convergence of our hybrid method.

Assumption 1 .

(i) The level set $\Omega = \{x \in R^n / f(x) \leq f(x_0)\}$ is bounded .
(ii) In some neighborhood N of Ω, the objective function f is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant $L > 0$ such that

$$\| g(x) - g(y) \| \leq L \| x - y \| \qquad \text{for all} \quad x, y \in N. \tag{26}$$

These assumptions imply that there exists a constant $\gamma \geq 0$ such that

$$\| g(x) \| \leq \gamma. \tag{27}$$

for all $x \in \Omega$ [30].

The next lemma gives the famous Zoutendijk conditions [31] proved in [32]

Lemma 1 *Suppose that the above assumptions (i) and (ii) hold and consider any iteration of the form* (2)*, where* d_k *satisfies the descent condition* $g_k^T d_k < 0$ *and* α_k *satisfies the Wolfe inexacte line search or its strong version. Then*

$$\sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < +\infty \tag{28}$$

Lemma 2 *[23] Suppose that the above assumptions (i) and (ii) hold. If* d_k *is a descent direction and the step length* α_k *satisfies*

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad \sigma < 1, \tag{29}$$

then

$$\alpha_k \geq \frac{1 - \sigma}{L} \frac{|d_k^T g_k|}{\| d_k \|^2}. \tag{30}$$

Proof Through the use of (29), the Cauchy Schwarz inequality and (26) it holds that:

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

Hence, the assertion (30) holds. From the second strong Wolfe condition (4) and the condition (21), α_k satisfies (30). According to the assumptions 1 and (21), it results $g_k^T d_k \neq 0$, $\forall\, k \geq 0$. Then, $\alpha_k = 0$ is not satisfying (4). This shows that α_k obtained in the HBGG method is not equal to zero, i.e. there exists a constant $\lambda > 0$ such that

$$\alpha_k \geq \lambda, \quad \text{for all} \quad k \geq 0. \tag{31}$$

□

The following theorem ensures the global convergence.

Theorem 3 *Suppose that Assumption 1 holds. Let the sequences $\{x_k\}$ be generated by HBGG algorithm. Then*

$$\lim_{k \mapsto \infty} inf \parallel g_k \parallel = 0. \tag{32}$$

Proof Suppose that (32) does not hold. Then, there exists a constant $r > 0$ such that

$$\parallel g_k \parallel > r. \tag{33}$$

From (8) we have

$$\begin{aligned} |\beta_k^{HBGG}| &\leq (1-\theta_k)|\beta_k^{DL}| + \theta_k|\beta_k^{DY}|, \\ &\leq |\beta_k^{DL}| + |\beta_k^{DY}|, \\ &= \frac{|g_{k+1}^T(y_k - ts_k)|}{|d_k^T y_k|} + \frac{\parallel g_{k+1} \parallel^2}{|d_k^T y_k|}, \\ &\leq \frac{|g_{k+1}^T y_k| + t|g_{k+1}^T s_k)|}{|d_k^T y_k|} + \frac{\parallel g_{k+1} \parallel^2}{|d_k^T y_k|}, \\ &\leq \frac{\parallel g_{k+1} \parallel \parallel y_k \parallel + t \parallel g_{k+1} \parallel \parallel s_k \parallel}{|d_k^T y_k|} + \frac{\parallel g_{k+1} \parallel^2}{|d_k^T y_k|}. \end{aligned} \tag{34}$$

According to the second strong Wolfe condition (4) and the sufficient descent condition (21) we have

$$d_k^T y_k \geq -(1-\sigma) g_k^T d_k \geq (1-\sigma) C \parallel g_k \parallel^2 .$$

From (33) we get:

$$d_k^T y_k \geq (1-\sigma) C r^2,$$

$$\frac{1}{d_k^T y_k} \leq \frac{1}{(1-\sigma) C r^2}. \tag{35}$$

From (26)

$$\parallel y_k \parallel = \parallel g_{k+1} - g_k \parallel \leq L \parallel x_{k+1} - x_k \parallel = L \parallel s_k \parallel \leq LD, \tag{36}$$

where, $D = \{\max \parallel x - y \parallel / x, y \in \Omega\}$ is the diameter of the level set Ω. From (27), (35) and (36) then (34) becomes

$$|\beta_k^{HBGG}| \leq \frac{\gamma LD + t\gamma D}{(1-\sigma)Cr^2} + \frac{\gamma^2}{(1-\sigma)Cr^2} = \frac{\gamma LD + t\gamma D + \gamma^2}{(1-\sigma)Cr^2} = E.$$

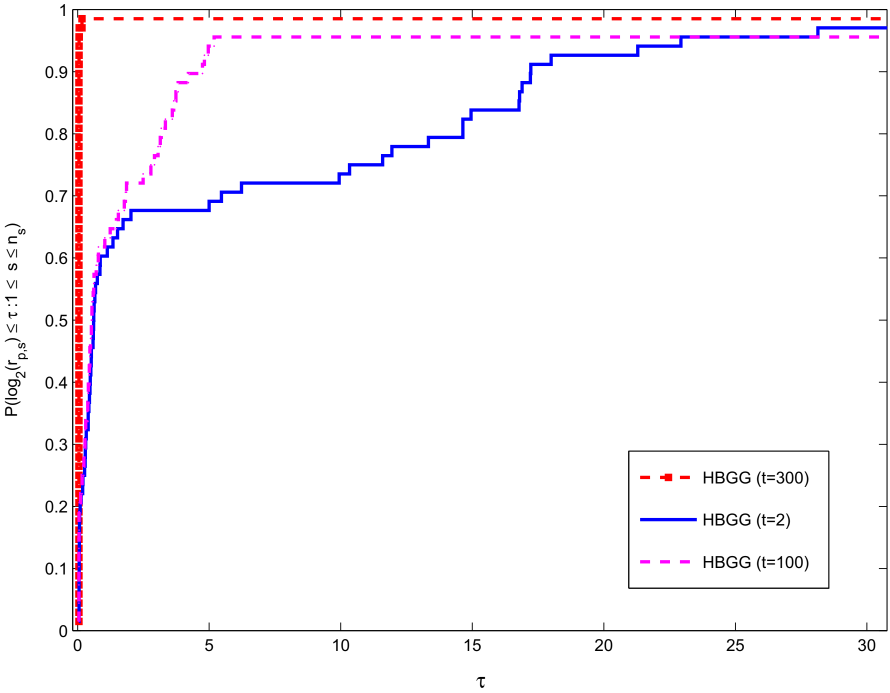

Fig. 1 Performance profile based on CPU time to choose the optimal t of HBGG algorithm

Therefore

$$\| d_{k+1} \| \leq \| g_{k+1} \| + |\beta_k^{HBGG}| \, \| d_k \| \leq \gamma + E \| d_k \| . \tag{37}$$

Using $\| d_k \| = \frac{\|s_k\|}{\alpha_k}$ and from (31), we get

$$\| d_{k+1} \| \leq \gamma + E \frac{\| s_k \|}{\alpha_k} \leq \gamma + E \frac{D}{\lambda} = M.$$

Now, we obtain

$$\sum_{k \geq 0} \frac{1}{\| d_k \|^2} = +\infty. \tag{38}$$

Moreover, from (28), (33) and (21), it results

$$C^2 r^4 \sum_{k \geq 0} \frac{1}{\| d_k \|^2} \leq \sum_{k \geq 0} \frac{C^2 \| g_k \|^4}{\| d_k \|^2} \leq \sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < +\infty, \tag{39}$$

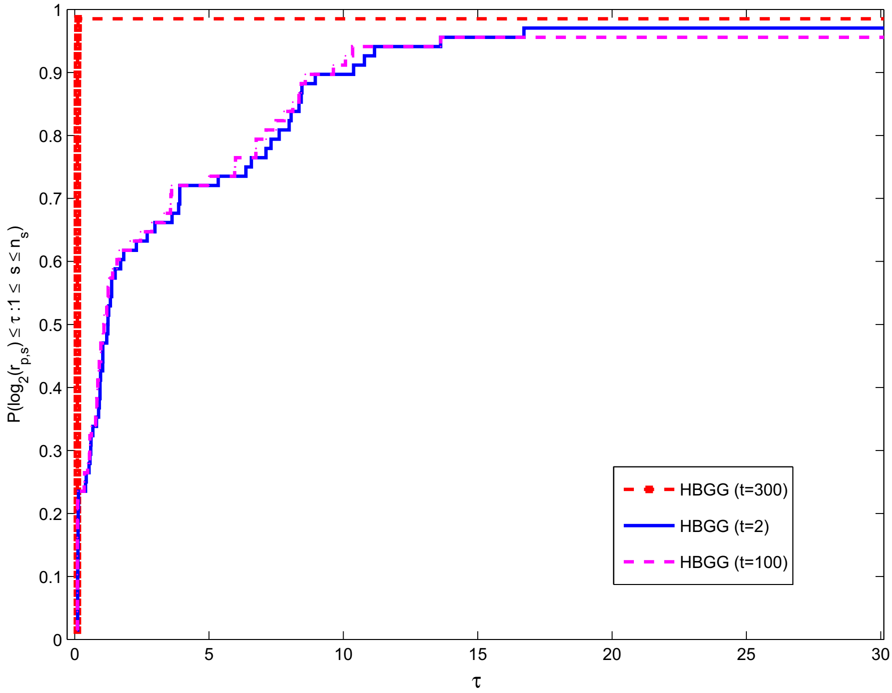

Fig. 2 Performance profile based on the number of iterations to choose the optimal t of HBGG algorithm

then

$$\sum_{k\geq 0} \frac{1}{\| d_k \|^2} < +\infty.$$

This is contradiction with (38), so we have proved (32) . □

4 Numerical experiment

In this section, we are going to discuss the numerical experiments of our HBGG algorithm by comparing it with those of DY [11] and DL [12] algorithms. For that we selected 80 unconstrained optimization test problems from [33], each problem is tested for this variables: 2, 50, 100, 200, 500, 1000, 2000, 3000, 5000 and 10000. All codes are compiled with a PC with the following specifications Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz, 4,00 Go RAM. We present the numericals comparisons with the other algorithms including the performance profiles given by Dolan and Moré [34], under strong Wolfe line search conditions (3), (4) with $\delta = 0.0001$ and $\sigma = 0.1$, and we use the stopping criterion $\| g_k \|_\infty \leq 10^{-7}$ for all algorithms.

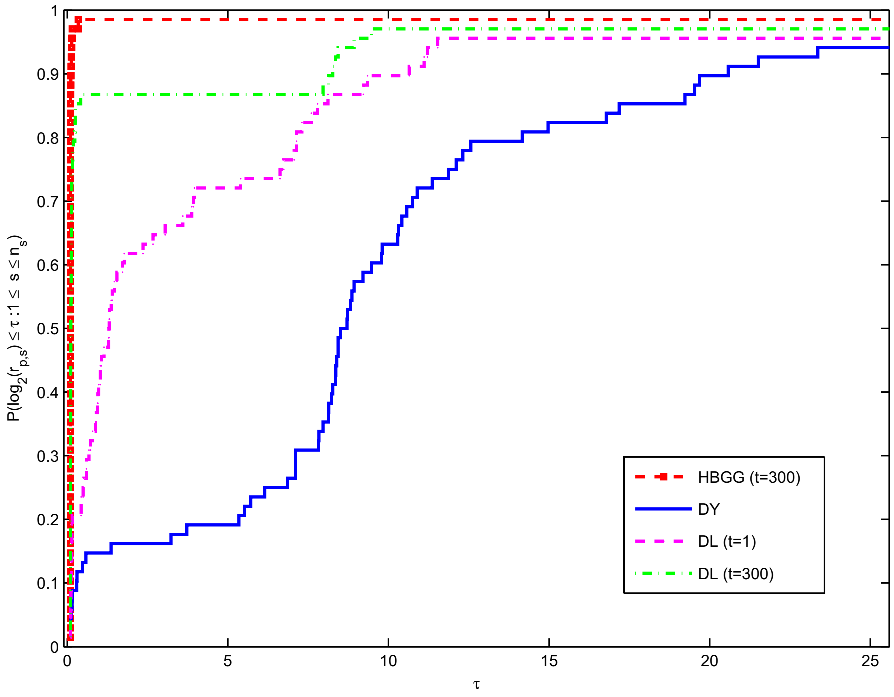

Fig. 3 Performance profile for CPU time

Firstly, we choose the best value of the parameter t. As Figures reffig:naimatempdydl and reffig:naimaiterdydl show, our HBGG algorithm with $t = 300$ performs better than $t = 2$ and $t = 100$ in terms of CPU time and the number of iterations.

Note that, concerning the DL method proposed by Dai and Liao [12] the numerical results were applied for t=1, so we compare HBGG algorithm for t=300 with DL algorithm for t=1 and t=300.

Figures 3, 4 and 5 show the performance profile based on CPU time, the number of iterations and the number of functions and gradient evaluations, respectively. All figures indicate that the performance of the HBGG algorithm for t=300 is significantly better than DY, DL(t=1) and DL(t=300) algorithms.

5 Conclusion

In this paper, we have presented a new hybrid conjugate gradient algorithm which has some good properties of the Newton method. In order to achieve this, we combined DY and DL methods as a new hybrid conjugate gradient method in which the parameter β_k is computed via a convex combination of β_k^{DY} and β_k^{DL}, ie: $\beta_k = (1-\theta_k)\beta_k^{DL} + \theta_k \beta_k^{DY}$, where the parameter θ_k was calculated in such a way that our search direction is equal to the Newton direction. Our algorithm does not need to calculate or store the the Hessian matrix needed by Newton's method. Using the strong Wolfe inexact line search we

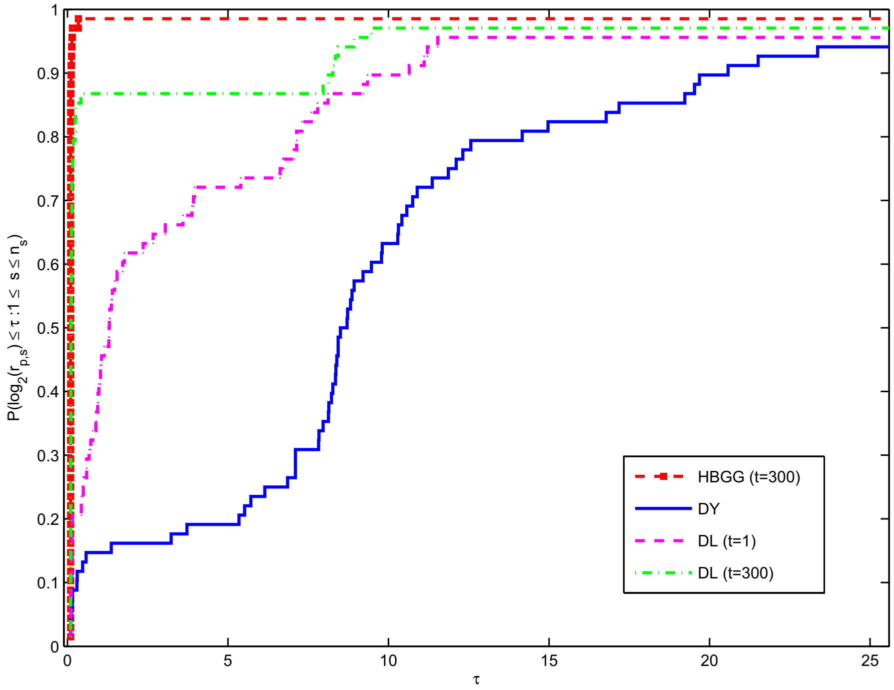

Fig. 4 Performance profile for the number of iterations

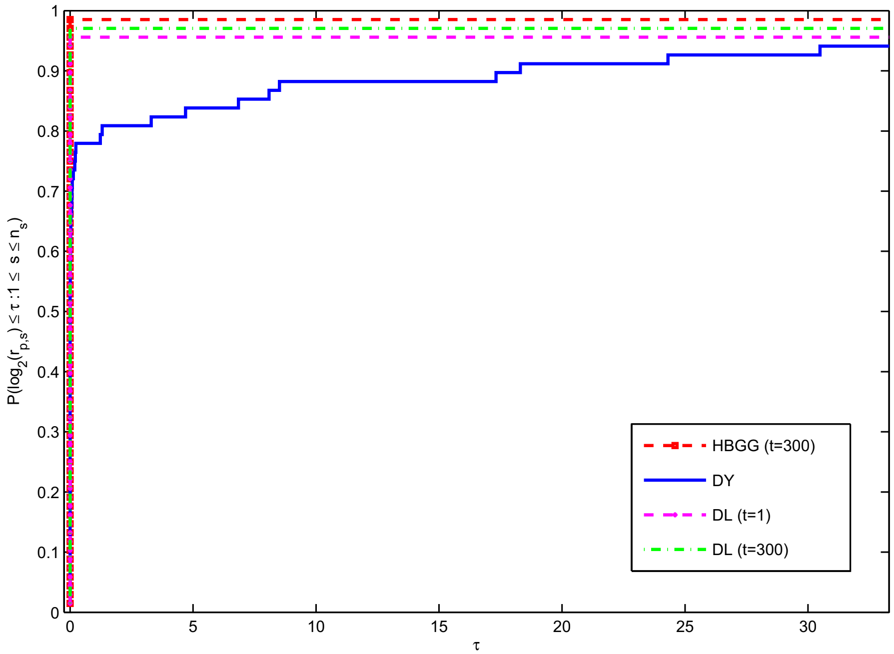

Fig. 5 Performance profile for functions and gradient evaluations

proved the sufficient descent property and global convergence. Our algorithm is more efficient than DY an DL algorithms, as shown by the numerical results In our future research, we will further develop our algorithm to be more efficient.

Acknowledgements The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions to improve this paper.

Data availability statement Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Babaie-Kafaki, S., Ghanbari, R.: A hybridization of the Hestenes–Stiefel and Dai–Yuan conjugate gradient methods based on a least-squares approach. Optim. Methods Softw. **30**(4), 673–681 (2015). https://doi.org/10.1080/10556788.2014.966825
2. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**(2), 226–235 (1969). https://doi.org/10.1137/1011036
3. Wolfe, P.: Convergence conditions for ascent methods. II: some corrections. SIAM Rev. **13**(2), 185–188 (1971). https://doi.org/10.1137/1013035
4. Narayanan, S., Kaelo, P.: A linear hybridization of Dai–Yuan and Hestenes–Stiefel conjugate gradient method for unconstrained optimization. Numer. Math. Theor. Meth. Appl. **14**, 527–539 (2021). https://doi.org/10.4208/nmtma.OA-2020-0056
5. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**(6), 409–436 (1952). https://doi.org/10.6028/jres.049.044
6. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**(2), 149–154 (1964). https://doi.org/10.1093/comjnl/7.2.149
7. Polak, E., Ribiére, G.: Note sur la convergence de méthodes de directions conjuguées. Rev. Fr. Inf. Rech. Oper. **3**(16), 35–43 (1969). https://doi.org/10.1051/m2an/196903R100351
8. Polyak, B.T.: The conjugate gradient method in extremal problems. USSR Comput. Math. Math. Phys. **9**(4), 94–112 (1969). https://doi.org/10.1016/0041-5553(69)90035-4
9. Fletcher, R.: Practical methods of optimization. Unconstrained Optimization. vol. 1, John Wiley and Sons, New York (1987)
10. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, part 1: theory. J. Optim. Theory Appl. **69**(1), 129–137 (1991). https://doi.org/10.1007/BF00940464
11. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999). https://doi.org/10.1137/S1052623497318992
12. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**(1), 87–101 (2001). https://doi.org/10.1007/s002450010019
13. Yao, S., Qin, B.: A hybrid of DL and WYL nonlinear conjugate gradient methods. Abstract and Applied Analysis, vol. 2014, Article ID 279891, p. 9 (2014). https://doi.org/10.1155/2014/279891
14. Xu, X., Kong, F.Y.: New hybrid conjugate gradient methods with the generalized Wolfe line search. SpringerPlus. **5**(1), 1–10 (2016). https://doi.org/10.1186/s40064-016-2522-9
15. Djordjević, S.S.: New hybrid conjugate gradient method as a convex combination of FR and PRP methods. Filomat. **30**(11), 3083–3100 (2016). https://doi.org/10.2298/FIL1611083D
16. Djordjević, S.S.: New hybrid conjugate gradient method as a convex combination of LS and CD methods. Filomat. **31**(6), 1813–1825 (2017). https://doi.org/10.2298/FIL1706813D
17. Kaelo, P., Narayanan, S., Thuto, M.V.: A modified quadratic hybridization of Polak-Ribiere-Polyak and Fletcher-Reeves conjugate gradient method for unconstrained optimization problems. Int. J. Opt. Control Theor.-Appl. (IJOCTA) **7**(2), 177–185 (2017). https://doi.org/10.11121/ijocta.01.2017.00339

18. Hassan, B.A., Owaid, O.A., Yasen, Z.T.: A variant of hybrid conjugate gradient methods based on the convex combination for optimization. Indones. J Electr. Eng. Comput. Sci. **20**(2), 1007–1015 (2020). https://doi.org/10.11591/ijeecs.v20.i2.pp1007-1015
19. Alhawarat, A., Salleh, D., Masmali, I.A.: A convex combination between two different search directions of conjugate gradient method and application in image restoration. Mathematical Problems in Engineering (2021). https://doi.org/10.1155/2021/9941757
20. Mohamed, N.S., Mamat, M., Rivaie, M., Shaharudin, S.M.: A comparison on classical-hybrid conjugate gradient method under exact line search. Int. J. Adv. Intell. Inform. **5**(2), 150–168 (2019). https://doi.org/10.26555/ijain.v5i2.356
21. Akinduko, O.B.: A new conjugate gradient method with sufficient descent property. Earthline J. Math. Sci. **6**(1), 163–174 (2021). https://doi.org/10.34198/ejms.6121.163174
22. Andrei, N.: Another hybrid conjugate gradient algorithm for unconstrained optimization. Numer. Algorithms **47**(2), 143–156 (2008). https://doi.org/10.1007/s11075-007-9152-9
23. Liu, J.K., Li, S.J.: New hybrid conjugate gradient method for unconstrained optimization. Appl. Math. Comput. **245**, 36–43 (2014). https://doi.org/10.1016/j.amc.2014.07.096
24. Jardow, F.N., Al-Naemi, G.M.: A new hybrid conjugate gradient algorithm as a convex combination of MMWU and RMIL nonlinear problems. J. Interdiscip. Math. **24**(3), 637–655 (2021). https://doi.org/10.1080/09720502.2020.1815346
25. 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). https://doi.org/10.1007/s10473-019-0117-6
26. Djordjević, S.S.: New hybrid conjugate gradient method as a convex combination of HS and FR conjugate gradient methods. J. Appl. Math. Comput. **2**(9), 366–378 (2018). https://doi.org/10.26855/jamc.2018.09.002
27. Abubakar, A.B., Kumam, P., Malik, M., Chaipunya, P., Ibrahim, A.H.: A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection. AIMS Math. **6**(6), 6506–6527 (2021). https://doi.org/10.3934/math.2021383
28. Li, M.: A three term Polak-Ribière-Polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method. J. Indus. Manag. Optim. **16**(1), 245–260 (2020). https://doi.org/10.3934/jimo.2018149
29. Livieris, I.E., Tampakas, V., Pintelas, P.: A descent hybrid conjugate gradient method based on the memoryless BFGS update. Numer. Algorithms **79**, 1169–1185 (2018). https://doi.org/10.1007/s11075-018-0479-1
30. Andrei, N.: A hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan algorithms. Stud. Inform. Control. **17**(4), 373–392 (2008)
31. Zoutendijk, G.: Nonlinear programming, computational methods. Integer and nonlinear programming. 37–86 (1970)
32. Dai, Y., Yuan, Y.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**(1), 33–47 (2001). https://doi.org/10.1023/A:1012930416777
33. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
34. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263