Numer Algor
https://doi.org/10.1007/s11075-018-0479-1

ORIGINAL PAPER

A descent hybrid conjugate gradient method based on the memoryless BFGS update

Ioannis E. Livieris[1] · **Vassilis Tampakas**[1] · **Panagiotis Pintelas**[2]

Received: 7 March 2017 / Accepted: 16 January 2018

Abstract In this work, we present a new hybrid conjugate gradient method based on the approach of the convex hybridization of the conjugate gradient update parameters of DY and HS+, adapting a quasi-Newton philosophy. The computation of the hybrization parameter is obtained by minimizing the distance between the hybrid conjugate gradient direction and the self-scaling memoryless BFGS direction. Furthermore, a significant property of our proposed method is that it ensures sufficient descent independent of the accuracy of the line search. The global convergence of the proposed method is established provided that the line search satisfies the Wolfe conditions. Our numerical experiments on a set of unconstrained optimization test problems from the CUTEr collection indicate that our proposed method is preferable and in general superior to classic conjugate gradient methods in terms of efficiency and robustness.

Keywords Unconstrained optimization · Conjugate gradient method · Frobenious norm · Self-scaled memoryless BFGS · Global convergence

✉ Ioannis E. Livieris
livieris@teiwest.gr

[1] Department of Computer Engineering & Informatics, Technological Educational Institute of Western Greece, GR 263-34, Patras, Greece

[2] Department of Mathematics, University of Patras, GR 265-00, Patras, Greece

Published online: 01 February 2018

1 Introduction

We consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and its gradient is denoted by $g(x) = \nabla f(x)$. Conjugate gradient methods is probably the most popular class of unconstrained optimization algorithms for engineers and mathematicians due to various applications in the industry and engineering fields [8, 21, 36, 38, 51, 56, 59]. This class of methods is characterized by their low memory requirements, simple computations, and strong global convergence properties. In general, a nonlinear conjugate gradient method generates a sequence of points $\{x_k\}$, starting from an initial point $x_0 \in \mathbb{R}^n$, using the recurrence

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots \tag{1.2}$$

where x_k is the k-th approximation to the solution of (1.1), $\alpha_k > 0$ is the stepsize obtained by a line search, and d_k is the search direction which is defined by

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0, \tag{1.3}$$

where $g_k = g(x_k)$. Conjugate gradient methods differ in their way of defining the update parameter β_k, since different choices of β_k give rise to distinct conjugate gradient methods with quite different computational efficiency and convergence properties. Hager and Zhang [31] presented an excellent survey in which they divided the essential CG methods in two main categories. The first category includes the Fletcher-Reeves (FR) method [28], the Dai-Yuan (DY) method [23], and the conjugate descent (CD) method [27] with the following update parameters:

$$\beta_k^{\mathrm{FR}} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \qquad \beta_k^{\mathrm{DY}} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \qquad \beta_k^{\mathrm{CD}} = -\frac{\|g_{k+1}\|^2}{d_k^T g_k},$$

which all share the common numerator $\|g_{k+1}\|^2$ in β_k. The second category includes the Polak-Ribière (PR) method [52], the Hestenes-Stiefel (HS) method [32], and the Liu and Storey (LS) method [37] which all have the same numerator $g_{k+1}^T y_k$ in β_k. The update parameters of these methods are respectively specified as follows:

$$\beta_k^{\mathrm{PR}} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \qquad \beta_k^{\mathrm{HS}} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \qquad \beta_k^{\mathrm{LS}} = -\frac{g_{k+1}^T y_k}{d_k^T g_k}.$$

The conjugate gradient methods in the first category possess strong global convergence properties [1, 24, 45] while the methods in the second category lack convergence in certain circumstances and as a result, they can cycle infinitely without presenting any substantial progress [55]. However, the methods in the first category usually exhibit poor computational performance due to the jamming phenomenon [54], i.e., the algorithms can take many short steps without making significant progress to the solution. In contrast, the methods in the second category possess an automatic approximate restart procedure which avoids jamming from occurring;

hence, their numerical performance is often superior to the performance of the methods with $\|g_{k+1}\|^2$ in the numerator of β_k.

In the literature, much effort has been devoted to develop new conjugate gradient methods which possess strong convergence properties and are also computationally superior to classical methods by hybridizing the above two approaches. The main idea behind the hybridization approach is to exploit the convergence properties of a conjugate gradient method from the first category and switch to a conjugate gradient from the second category when the iterations jam. Along this line, sample works include the hybridizations of FR and PR methods [16, 29, 33, 58], the hybridizations of HS and DY methods [25, 60], and the hybridization of LS and CD methods [58]. Notice that, in these methods, the update parameter is calculated based on discrete combinations of update parameters of the two categories.

Recently, Andrei [8–10] proposed a new class of hybrid conjugate gradient algorithms which is based on the concept of convex combination of classical conjugate gradient algorithms. Generally, the performance of the hybrid variants based on the concept of convex combination is better than that of the constituents. In recent efforts following Andrei's approach, Babaie-Kafaki et al. [12, 13, 15, 19] proposed some globally convergent conjugate gradient methods (HCG+) in which the update parameter β_k is determined as the convex combination of β_k^{DY} and $\beta_k^{\mathrm{HS+}}$, namely

$$\beta_k^{\mathrm{HCG+}} = \lambda_k \beta_k^{\mathrm{DY}} + (1 - \lambda_k)\beta_k^{\mathrm{HS+}}, \tag{1.4}$$

with $\beta_k^{\mathrm{HS+}} = \max\{\beta_k^{\mathrm{HS}}, 0\}$ and the scalar $\lambda_k \in [0, 1]$ is the hybridization parameter. Notice that if $\lambda_k = 0$, then $\beta_k^{\mathrm{HCG+}} = \beta_k^{\mathrm{HS+}}$ and if $\lambda_k = 1$, then $\beta_k^{\mathrm{HCG+}} = \beta_k^{\mathrm{DY}}$. Based on their numerical experiments, the authors concluded that the computational performance of the HCG+ method is heavily dependent on the choice of the hybridization parameter λ_k [13]. Moreover, in order to enhance the performance of their proposed method, the hybridization parameter is adaptively calculated by

$$\lambda_k = -2\frac{\|y_k\|^2}{s_k^T yk}\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}, \tag{1.5}$$

based on the study of a modified secant equation.

In this work, we present a new hybrid conjugate gradient method based on the approach of the convex hybridization of the conjugate gradient update parameters of DY and HS+, adapting a quasi-Newton philosophy. More specifically, the value of the hybrization parameter is obtained by minimizing the distance between the hybrid conjugate gradient direction and the self-scaling memoryless BFGS direction. Additionally, an attractive property of our proposed method is that it ensures sufficient descent independent of the accuracy of the line search. The global convergence of our proposed method is established, under the Wolfe lines search conditions.

The remainder of this paper is organized as follows. In Section 2, we present a brief discussion on the self-scaling memoryless BFGS method and in Section 3, we introduce our new hybrid conjugate gradient method. In Section 4, we present the global convergence analysis. Section 5 reports our numerical experiments on a

set of unconstrained optimization test problems from the CUTEr collection [20] utilizing the performance profiles of Dolan and Morè. Finally, Section 6 presents our concluding remarks.

2 Self-scaling memoryless BFGS

The self-scaling memoryless BFGS method is generally considered as one of the most efficient method for solving large-scale optimization problems [11, 34, 45, 62] due to its strong theoretical properties and favorable computational performance. Moreover, it provides a good understanding about the relationship between nonlinear conjugate gradient methods and quasi-Newton methods [7, 50, 57].

Generally, the self-scaled memoryless BFGS matrices are computed based on the L-BFGS philosophy [35, 43] using information from the most recent iteration. Given an initial matrix $B_0 = \theta_0 I$ with $\theta_0 \in \mathbb{R}^*$, and the BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k},$$

the resulting scaled memoryless BFGS update scheme takes the form

$$B_{k+1} = \theta_k I - \theta_k \frac{s_k s_k^T}{s_k^T s_k} + \frac{y_k y_k^T}{s_k^T y_k}.$$

where $\theta_k \in \mathbb{R}$ is the scaling parameter. Additionally, the search direction in this method is generated by

$$d_{k+1} = -B_{k+1}^{-1} g_{k+1},$$

where B_{k+1}^{-1} is the inverse of Hessian approximation which can be easily calculated by the following expression [43]:

$$B_{k+1}^{-1} = \frac{1}{\theta_k} I - \frac{1}{\theta_k} \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + \left(1 + \frac{1}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{s_k s_k^T}{s_k^T y_k}, \tag{2.1}$$

It is pointed out by many researchers that efficiency of the self-scaled memoryless BFGS is heavily depended of the selection of the scaling parameter θ_k. The idea behind scaling is to achieve an ideal distribution of the eigenvalues of update formulae (2.1), improving its condition number and consequently increasing the numerical stability of the method [45]. Based on the analysis of quadratic objective functions, there have been proposed two very popular and effective adaptive formulas for the computation of θ_k. The first one have been proposed by Oren and Luenberger [48]

$$\theta_k^{\mathrm{OL}} = \frac{s_k^T y_k}{\|s_k\|^2}, \tag{2.2}$$

while the second one by Oren and Spedicato [49]

$$\theta_k^{\mathrm{OS}} = \frac{\|y_k\|^2}{s_k^T y_k}. \tag{2.3}$$

However, Nocedal and Yuan [46] reported some very disappointing numerical experiments in which the best self-scaling BFGS algorithm of Oren and Luenberger [48] performs badly compared to the classical BFGS algorithm when applied with inexact line search to a simple quadratic function of two variables. To address this problem, Al-Baali [2] proposed the condition

$$\theta_k \leq 1 \tag{2.4}$$

and presented a globally and superlinearly convergent BFGS method with inexact line search. The motivation behind condition (2.4) is based on the fact that the eigenvalues of Hessian approximation B_{k+1} can be reduced if $\theta_k < 1$, and hence, smaller eigenvalues are introduced in B_{k+1} if the eigenvalues of B_k are large. Moreover, the BFGS update formula has the significant property of self-correcting the small eigenvalues [4, 44, 45, 53]. Thus, condition (2.4) ensures keeping the eigenvalues of the Hessian approximation matrix within a suitable range, and as a result, if B_k incorrectly approximates the curvature of the objective function and this estimate slows down the iteration, then the next Hessian approximation will tend to correct itself in the next few steps. Numerical evidences [2, 3] show that the performance of the self-scaling BFGS was improved substantially and concluded that the proposed scaled method was computationally superior to the original one. For more choices and information on scalar θ_k, we refer to [2, 3, 5, 46–49] and the references therein.

Independently, another interesting approach was proposed by Zou et al. [62] for studying the computational performance of several limited-memory quasi-Newton and truncated Newton methods. In particular, they performed comparative tests on several synthetic function problems allowing control of the clustering of eigenvalues in the Hessian spectrum. In this way, they examined each method's sensitivity to various degrees of ill conditioning and evaluated its computational performance as the condition number increases.

3 An adaptive descent hybrid conjugate gradient method

Motivated by the computational efficiency of the self-scaling memoryless BFGS, we propose an adaptive choice for parameter λ_k in (1.4), following a similar methodology of that in [18, 22]. More specifically, we define parameter λ_k in such a way to reduce the distance between the search direction matrix of the HCG+ and the self-scaled memoryless BFGS update.

For this purpose, following Perry's point of view, it is notable that from (1.3) and (1.4), the search direction of the HCG+ method can be written as

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{3.1}$$

where

$$Q_{k+1} = I - \lambda_k \frac{d_k g_{k+1}^T}{d_k^T y_k} - (1 - \lambda_k) \frac{d_k y_k^T}{d_k^T y_k}.$$

Therefore, the HCG+ method can be considered as a quasi-Newton method [24, 45] in which the inverse Hessian is approximated by the nonsymmetric matrix Q_{k+1}.

Subsequently, based on the above discussion, we compute parameter λ_k as the solution of the following minimization problem

$$\min_{\lambda_k>0} \|D_{k+1}\|_F \tag{3.2}$$

where $D_{k+1} = Q_{k+1}^T - B_{k+1}^{-1}$ and $\|\cdot\|_F$ is the Frobenius matrix norm. Since $\|D_{k+1}\|_F^2 = \mathrm{tr}(D_{k+1}^T D_{k+1})$ and after some algebra, we obtain

$$\|D_{k+1}\|_F^2 = \lambda_k^2 \frac{\|s_k\|^2\|g_k\|^2}{(s_k^T y_k)^2} - 2\lambda_k \left[\frac{s_k^T g_k}{s_k^T y_k} + \left(\frac{1}{\theta_k} - 1\right) \frac{\|s_k\|^2 (y_k^T g_k)}{(s_k^T y_k)^2} - \left(1 + \frac{1}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{\|s_k\|^2 (s_k^T g_k)}{(s_k^T y_k)^2} \right] + \xi,$$

where ξ is a real constant, independent of λ_k. Clearly, the computation of $\|D_{k+1}\|_F^2$ can be considered as a second-degree polynomial of variable λ_k where the coefficient of λ_k^2 is always positive. Therefore, the unique solution of the minimization problem (3.2) is given by

$$\lambda_k^* = \frac{s_k^T g_k}{\|g_k\|^2} \left[\frac{s_k^T y_k}{\|s_k\|^2} - \frac{1}{\theta_k} \frac{\|y_k\|^2}{s_k^T y_k} - 1 \right] + \left(\frac{1}{\theta_k} - 1\right) \frac{y_k^T g_k}{\|g_k\|^2} \tag{3.3}$$

Clearly, an important property of the value of λ_k^* is that matrix Q_{k+1} is as close as possible to the self-scaling memoryless BFGS matrix. Moreover, in order to have a convex combination in (1.4), we restrict the values of λ_k in the interval [0, 1], namely if $\lambda_k^* < 0$ then we set $\lambda_k^* = 0$ and also, if $\lambda_k^* > 1$, then we set $\lambda_k^* = 1$.

In order to guarantee that our proposed method generates descent directions and increase further its computational efficiency and robustness, we exploit the idea of the modified FR method [61]. More specifically, let the search direction be defined by

$$d_{k+1} = -\left(1 + \beta_k^{\mathrm{HCG+}} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|}\right) g_{k+1} + \beta_k^{\mathrm{HCG+}} d_k. \tag{3.4}$$

It is easy to see that the condition holds, using any line search

$$d_{k+1}^T g_{k+1} \leq -\|g_{k+1}\|^2. \tag{3.5}$$

At this point, we present our adaptive descent hybrid conjugate gradient algorithm (ADHCG).

4 Convergence analysis

In this section, we present the global convergence analysis of algorithm ADHCG, under the following assumptions on the objective function f.

Assumption 1 The level set $\mathscr{L} = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded; namely, there exists a positive constant B, such that

$$\|x\| \leq B, \quad \forall x \in \mathscr{L}. \tag{4.1}$$

Algorithm 1 (ADHCG)

Step 1: Initiate $x_0 \in \mathbb{R}^n$ and $0 < \sigma_1 < \sigma_2 < 1$; Set $k = 0$.
Step 2: If $\|g_k\| = 0$, then terminate; Otherwise go to the next step.
Step 3: Compute the descent direction d_k by (1.4), (3.3) and (3.4).
Step 4: Determine a stepsize α_k using the Wolfe line search:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \sigma_1 \alpha_k g_k^T d_k, \tag{3.6}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k. \tag{3.7}$$

Step 5: Let $x_{k+1} = x_k + \alpha_k d_k$.
Step 6: Set $k = k + 1$ and go to Step 2.

Assumption 2 In some neighborhood $\mathcal{N}$ of $\mathcal{L}$, f is differentiable and its gradient g is Lipschitz continuous, i.e., there exists a positive constant L, such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{N}. \tag{4.2}$$

Since $\{f_k\}$ is a decreasing sequence, it is clear that the sequence $\{x_k\}$ generated by Algorithm ADHCG is contained in $\mathcal{L}$ and there exists a constant f^*, such that

$$\lim_{k\to\infty} f(x_k) = f^*.$$

Furthermore, it follows directly from Assumptions 1 and 2 that there exists a positive constant $M > 0$ such that

$$\|g(x)\| \leq M, \quad \forall x \in \mathcal{L}. \tag{4.3}$$

In order to present the convergence analysis, the following lemma is needed which constitutes a general result of conjugate gradient methods implemented with a line search that satisfies the Wolfe line search conditions (3.6) and (3.7).

Lemma 4.1 *Suppose that Assumptions 1 and 2 hold. Consider any method of the form* (1.2) *where d_k is a descent direction, i.e., $d_k^T g_k < 0$ and α_k satisfies the Wolfe conditions* (3.6) *and* (3.7)*, then*

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

Obviously, it immediate follows from Lemma 4.1 and (3.5), that

$$\sum_{k\geq 0} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty, \tag{4.4}$$

which is very useful for the global convergence analysis.

Subsequently, we show that the Algorithm ADHCG is globally convergent for general nonlinear functions. For this purpose, we present some properties for the search direction d_k, formula $\beta_k^{\mathrm{HCG+}}$, and step s_k. In the rest of this section, we

assume that the sequence $\{\theta_k\}$ is uniformly bounded; namely, there exist positive constants $\theta_{\min}$ and $\theta_{\max}$ such that

$$\theta_{\min} \leq \theta_k \leq \theta_{\max} \tag{4.5}$$

Lemma 4.2 *Suppose that Assumptions 1 and 2 hold. Let $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm ADHCG, if there exists a constant $\mu > 0$ such that*

$$\|g_k\| \geq \mu, \quad \forall k \geq 0, \tag{4.6}$$

and then there exist positive constants C_1 and C_2 such that for all $k \geq 1$

$$|\beta_k^{\text{HCG+}}| \leq C_1 \|s_k\| \tag{4.7}$$

and

$$|\beta_k^{\text{HCG+}}| \frac{|g_{k+1}^T d_k|}{\|g_{k+1}\|^2} \leq C_2 \|s_k\|. \tag{4.8}$$

Proof Firstly, we show that there exists a constant $\eta > 0$ such that

$$\lambda_k \leq \eta \|s_k\|. \tag{4.9}$$

From (3.5) and (3.7), we have

$$d_k^T y_k \geq (\sigma_2 - 1) g_k^T d_k = (1 - \sigma_2)\|g_k\|^2. \tag{4.10}$$

Combining this with Assumptions 1 and 2 and relations (3.3), (3.5), (4.3), (4.5), and (4.6), we obtain

$$\begin{aligned} |\lambda_k| &\leq \frac{|s_k^T g_k|}{\|g_k\|^2} \left[\frac{|s_k^T y_k|}{\|s_k\|^2} + \frac{1}{|\theta_k|} \frac{\|y_k\|^2}{|s_k^T y_k|} + 1 \right] + \left(\frac{1}{|\theta_k|} + 1 \right) \frac{|y_k^T g_k|}{\|g_k\|^2} \\ &\leq \frac{\|y_k\|}{\|g_k\|} + \frac{\|y_k\|^2}{|\theta_k|(1-\sigma_2)\|g_k\|^2} + \frac{\|s_k\|}{\|g_k\|} + \left(\frac{1}{|\theta_k|} + 1 \right) \frac{\|y_k\|}{\|g_k\|} \\ &\leq \left[\frac{L^2 B}{\theta_{\min}(1-\sigma_2)\mu^2} + \frac{1}{\mu} \left(\frac{1}{\theta_{\min}} + L + 2 \right) \right] \|s_k\| \end{aligned} \tag{4.11}$$

Letting $\eta = \frac{L^2 B}{\theta_{\min}(1-\sigma_2)\mu^2} + \frac{1}{\mu}\left(\frac{1}{\theta_{\min}} + L + 2\right)$ then (4.9) is satisfied. Moreover, utilizing (1.4), (4.3), (4.6), (4.9), and (4.10), we get

$$|\beta_k^{\text{HCG+}}| \leq \frac{|g_{k+1}^T y_k|}{d_k^T y_k} + |\lambda_k| \frac{\|g_{k+1}\|^2}{d_k^T y_k} = \frac{ML + \eta M^2}{(1-\sigma_2)\mu^2} \|s_k\|.$$

Therefore, if we let $C_1 = \frac{ML + \eta M^2}{(1-\sigma_2)\mu^2}$, then (4.7) holds. Furthermore, by the Wolfe condition (3.7), we have

$$g_{k+1}^T d_k \geq \sigma_2 g_k^T d_k \geq -\sigma_2 y_k^T d_k + \sigma_2 g_{k+1}^T d_k. \tag{4.12}$$

Also, observe that

$$g_{k+1}^T d_k = y_k^T d_k + g_k^T d_k \leq d_k^T y_k. \tag{4.13}$$

By rearranging the inequality (4.12), we obtain $g_{k+1}^T d_k \geq -(\sigma_2/1-\sigma_2) d_k^T y_k$, which together with (4.13), we get

$$\left| \frac{g_{k+1}^T d_k}{d_k^T y_k} \right| \leq \max \left\{ \frac{\sigma_2}{(1-\sigma_2)}, 1 \right\}. \tag{4.14}$$

It immediate follows from Assumption 2 and relations (4.6), (4.9), and (4.14)

$$|\beta_k^{\text{HCG+}}| \frac{|g_{k+1}^T d_k|}{\|g_{k+1}\|^2} \leq \left(\frac{\|y_k\|}{\|g_{k+1}\|} + |\lambda_k| \right) \left| \frac{g_{k+1}^T d_k}{d_k^T y_k} \right| \leq \left(\frac{L}{\mu} + \eta \right) \max \left\{ \frac{\sigma_2}{(1-\sigma_2)}, 1 \right\} \|s_k\|$$

Letting $C_2 = \left(\frac{L}{\mu} + \eta \right) \max \left\{ \frac{\sigma_2}{(1-\sigma_2)}, 1 \right\}$, we obtain (4.8) which completes the proof. □

Subsequently, we present a lemma which shows that, asymptotically, the search directions change slowly.

Lemma 4.3 *Suppose that Assumptions 1 and 2 hold. Let* $\{x_k\}$ *and* $\{d_k\}$ *be generated by Algorithm ADHCG, if there exists a constant* $\mu > 0$*, such that (4.6) holds; then* $d_k \neq 0$ *and*

$$\sum_{k \geq 0} \|w_{k+1} - w_k\|^2 < \infty, \tag{4.15}$$

where $w_k = d_k / \|d_k\|$.

Proof Firstly, note that $d_k \neq 0$, for otherwise (3.5) would imply $g_k = 0$. Therefore, w_k is well defined. Let us define

$$r_{k+1} := \frac{\upsilon_{k+1}}{\|d_{k+1}\|} \quad \text{and} \quad \delta_{k+1} := \beta_k^{\text{HCG+}} \frac{\|d_k\|}{\|d_{k+1}\|}, \tag{4.16}$$

where

$$\upsilon_{k+1} = -\left(1 + \beta_k^{\text{HCG+}} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2} \right) g_{k+1}.$$

Then, by (3.4), we have

$$w_{k+1} = r_{k+1} + \delta_{k+1} w_k. \tag{4.17}$$

Using this relation with the identity $\|w_{k+1}\| = \|w_k\| = 1$ yields

$$\|r_{k+1}\| = \|w_{k+1} - \delta_{k+1} w_k\| = \|w_k - \delta_{k+1} w_{k+1}\|.$$

Moreover, using this with the fact that $\delta_{k+1} \geq 0$, we obtain

$$\|w_{k+1} - w_k\| \leq \|w_{k+1} - \delta_{k+1} w_k\| + \|w_k - \delta_{k+1} w_{k+1}\| = 2\|r_{k+1}\|.$$

Subsequently, we estimate an upper bound for $\|\upsilon_{k+1}\|$. It immediate follows from the definition of υ_{k+1} in (4.17) and relations (4.1), (4.3), and (4.10) that there exists a constant $D > 0$ such that

$$\|\upsilon_{k+1}\| \leq \left\| \left(1 + |\beta_k^{\text{HCG+}}| \frac{|g_{k+1}^T d_k|}{\|g_{k+1}\|^2} \right) g_{k+1} \right\| \leq (1 + C_2 B) M \triangleq D.$$

Thus, we have established an upper bound for $\|\upsilon_{k+1}\|$. Therefore, utilizing the previous relation with (4.4), (4.6), and (4.16), we obtain

$$\sum_{k\geq 0}\|w_{k+1}-w_k\|^2 = 4\sum_{k\geq 0}\|r_{k+1}\|^2 \leq 4\sum_{k\geq 0}\frac{\|\upsilon_{k+1}\|^2}{\|d_{k+1}\|^2} = 4\frac{D^2}{\mu^4}\sum_{k\geq 0}\frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2} < +\infty,$$

which completes the proof. □

Next, utilizing Lemmas 4.2 and 4.3, we establish the global convergence theorem for Algorithm ADHCG whose proof is similar to that of Theorem 3.2 in [30]; however, we present it here for completeness.

Theorem 4.1 *Suppose that Assumptions 1 and 2 hold. If $\{x_k\}$ is obtained by Algorithm ADCGH+, then we have*

$$\lim_{k\to\infty}\inf\|g_k\| = 0. \tag{4.18}$$

Proof We proceed by contraction, we suppose that the conclusion (4.18) is not true. That is, there exists a constant $\mu > 0$ such that for all k, $\|g_k\| \geq \mu$. The proof is divided in the following steps:

Step I. A bound on the step s_k. Observe that for any $l \geq k$, we have

$$x_l - x_k = \sum_{j=k}^{l-1}(x_{j+1}-x_j) = \sum_{j=k}^{l-1}\|s_j\|w_j = \sum_{j=k}^{l-1}\|s_j\|w_k + \sum_{j=k}^{l-1}\|s_j\|(w_j-w_k).$$

Utilizing Assumption 1 with the triangle inequality, we obtain

$$\sum_{j=k}^{l-1}\|s_j\| \leq \|x_l - x_k\| + \sum_{j=k}^{l-1}\|s_j\|\,\|w_j - w_k\| \leq B + \sum_{j=k}^{l-1}\|s_j\|\,\|w_j-w_k\|. \tag{4.19}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \geq 4BC_1, \tag{4.20}$$

where B and C are defined in (4.1) and (4.7), respectively. By Lemma 4.3, we can choose k_0 large enough such that

$$\sum_{i\geq k_0}\|w_{i+1}-w_i\|^2 \leq \frac{1}{4\Delta}. \tag{4.21}$$

For any $j > k \geq k_0$ with $j - k \leq \Delta$, using (4.21) with the Cauchy-Schwartz inequality, we obtain

$$\|w_j - w_k\| \leq \sum_{i=k}^{j-1}\|w_{i+1}-w_i\| \leq \sqrt{j-k}\left(\sum_{i=k}^{j-1}\|w_{i+1}-w_i\|\right)^{1/2} \leq \sqrt{\Delta}\left(\frac{1}{4\Delta}\right)^{1/2} = \frac{1}{2}.$$

Using this with (4.19), yields

$$\sum_{j=k}^{l-1} \|s_j\| < 2B, \tag{4.22}$$

where $l > k > k_0$ and $l - k \geq \Delta$.

Step II. A bound on the search directions d_l. We rewrite (3.4) as follows:

$$d_l = -g_l + \beta_k^{\text{MP+}} \left(I - \frac{g_l g_l^T}{\|g_l\|^2} \right) d_{l-1}. \tag{4.23}$$

Since g_l is orthogonal to $\left(I - \frac{g_l g_l^T}{\|g_l\|^2} \right) d_{l-1}$ and $I - \frac{g_l g_l^T}{\|g_l\|^2}$ is a project matrix, we have from (4.3), (4.7), and (4.23) that

$$\|d_l\|^2 \leq (\|g_l\| + |\beta_k^{\text{HCG+}}| \, \|d_{l-1}\|)^2 \leq 2\|g_l\|^2 + 2|\beta_k^{\text{HCG+}}|^2 \|d_{l-1}\|^2 \leq 2M^2 + 2C_1^2 \|s_{l-1}\|^2 \|d_{l-1}\|^2.$$

Defining $S_i = 2C_1^2 \|s_i\|^2$, we have that for $l > k_0$,

$$\|d_l\|^2 \leq 2M^2 \left(\sum_{i=k_0+1}^{l} \prod_{j=i}^{l-1} S_j \right) + \|d_{k_0}\|^2 \prod_{j=k_0}^{l-1} S_j. \tag{4.24}$$

Above, the product is defined to be 1 whenever the index range is vacuous. Next, let us consider as follows a product of Δ consecutive S_j, where $k \geq k_0$. Utilizing (4.20) and (4.22) together with the Cauchy-Schwartz inequality, we have

$$\prod_{j=k}^{k+\Delta-1} S_j = \prod_{j=k}^{k+\Delta-1} 2C_1^2 \|s_j\|^2 \leq \left(\frac{\sum_{j=k}^{k+\Delta-1} \sqrt{2} C_1 \|s_j\|}{\Delta} \right)^{2\Delta} \leq \left(\frac{2\sqrt{2} B C_1}{\Delta} \right)^{2\Delta} \leq \frac{1}{2^{\Delta}}.$$

Since the product of Δ consecutive S_j is bounded by $1/2^{\Delta}$, it immediate follows from (4.24) that $\|d_l\|^2 \leq c_1 l + c_2$ for a certain constant $c_1 > 0$ independent of l. Therefore, we have

$$\sum_{k \geq 0} \frac{\|g_k\|^4}{\|d_k\|^2} \geq \sum_{k \geq 0} \frac{\mu^2}{c_1 k + c_2} = +\infty,$$

which contradicts with (4.4). This completes the proof. □

5 Experimental results

In this section, we report some numerical results in order to evaluate the performance of our proposed conjugate gradient method ADHCG with that of the CG-DESCENT method [30], hybrid-enriched method [40], and the HCG+ method [13].

We selected 134 problems from the CUTEr [20] library which have been also tested in [13, 30, 39]. The implementation code was written in C and compiled with `gcc` (with compiler settings `-O3 -lm -c`) on a PC (2.66-GHz Quad-Core processor, 4 Gbyte RAM) running Linux operating system. The CG-DESCENT code

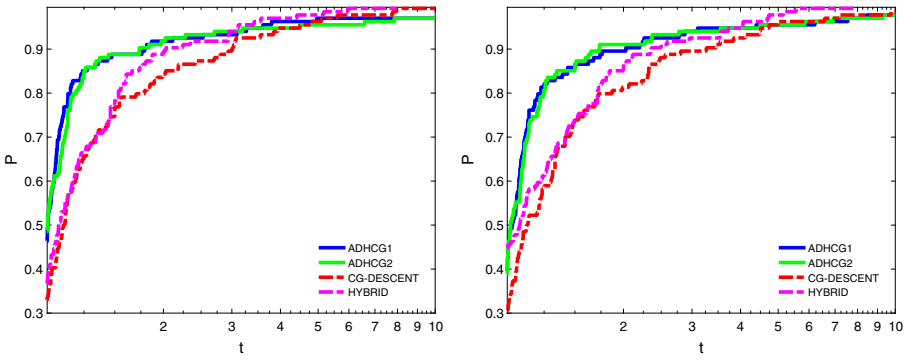
is coauthored by Hager and Zhang obtained from Hager's web page[1]. The hybrid-enriched method consists of interlacing in a dynamical way the L-BFGS method [35] with the TN method [41, 42] in order to explore the advantages of both of them. More specifically, in this method, l steps of the L-BFGS method are alternated with t steps of the T-N method. In our experiments, we set $l = 5$ and $t = 20$ as in [6]. The detailed numerical results can be found in http://www.math.upatras.gr/~livieris/Results/ADHCG.zip. In our experiments, we use the condition $\|g_k\|_\infty \leq 10^{-6}$ as stopping criterion and all algorithms were implemented with the same line search presented in [30].

All algorithms were evaluated using the performance profiles proposed by Dolan and Morè [26] relative to function evaluations, gradient evaluations, number of iterations, and CPU time (in seconds). The use of profiles provide a wealth of information such as solver efficiency, robustness, and probability of success in compact form and eliminate the influence of a small number of problems on the benchmarking process and the sensitivity of results associated with the ranking of solvers [26]. The performance profile plots the fraction P of problems for which any given method is within a factor τ of the best solver. The horizontal axis of the figure gives the percentage of the test problems for which a method is the fastest (efficiency), while the vertical axis gives the percentage of the test problems that were successfully solved by each method (robustness). The curves in the following figures have the following meaning:

- "ADHCG$_1$" stands for Algorithm ADHCG in which the scaling parameter is defined by $\theta_k = \min\left\{\theta_k^{\mathrm{OL}}, 1\right\}$.
- "ADHCG$_2$" stands for Algorithm ADHCG in which the scaling parameter is defined by $\theta_k = \min\left\{\theta_k^{\mathrm{OS}}, 1\right\}$.
- "CG-DESCENT" stands for the CG-DESCENT method (version 5.3) [30].
- "HYBRID" stands for the hybrid-enriched method [40].
- "HCG+" stands for the CG method with the update parameter $\beta_k^{\mathrm{HCG+}}$ in which λ_k is defined by (1.5) [13].

Figure 1 presents the performance profiles of ADHCG$_1$, ADHCG$_2$, CG-DESCENT, and HYBRID based on number of function evaluations and number of gradient evaluations. Clearly, our proposed methods outperform the classical methods CG-DESCENT and HYBRID with ADHCG$_2$ presenting slightly better performance, relative to both performance metrics. More analytically, the performance profile for function evaluations reports that ADHCG$_1$ and ADHCG$_2$ solve about 40.6 and 44.8% of the test problems with the least number of function evaluations, respectively, while CG-DESCENT and HYBRID solve about 31.3 and 35% of the test problems, respectively. Moreover, Fig. 1b illustrates that HYBRID is the most robust method, since it solves 42.4% of the test problems with the least number of gradient evaluations, while both our proposed methods solve about 35.1% the test problems. However, ADHCG$_1$ and ADHCG$_2$ are the most efficient methods, since their curves lie on the top.

[1] http://clas.ufl.edu/users/hager/papers/Software/

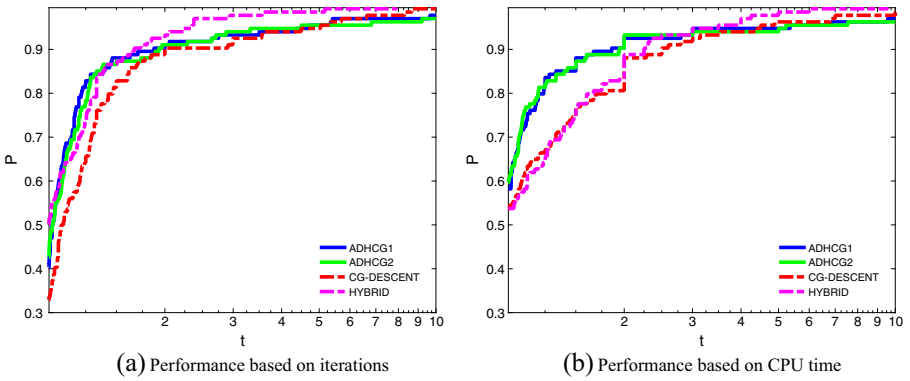


(a) Performance based on function evaluations (b) Performance based on gradient evaluations

Fig. 1 Log_{10} scaled performance profiles for ADHCG_1, ADHCG_2, CG-DESCENT, and HYBRID based on number of function evaluations (**a**) and number of gradient evaluations (**b**)

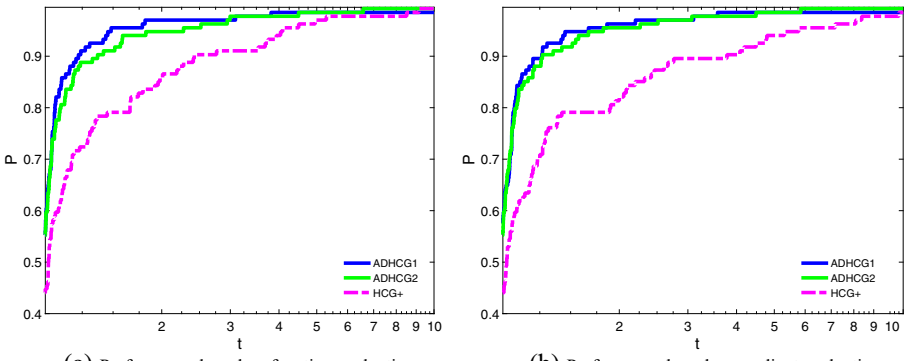
Figure 2 presents the performance profiles comparing ADHCG_1, ADHCG_2, CG-DESCENT, and HYBRID based on number of iterations and CPU time. As regards the number of iterations, HYBRID illustrates the highest probability of being the optimal solver since it corresponds to the top curve, slightly outperforming our proposed methods. The interpretation of Fig. 2b shows that ADHCG_1 and ADHCG_2 exhibit the best performance with respect to CPU time since they solve 76 and 80 out of 134 test problems with the least computational time, respectively, while CG-DESCENT and HYBRID solve only 72 of the test problems. Based on the above observations, we conclude that both our proposed methods outperform CG-DESCENT and HYBRID, in terms of efficiency and efficacy, regarding all performance metrics.

Figures 3 and 4 present the performance profiles of ADHCG_1, ADHCG_2, and HCG+, relative to all performance metrics. Obviously, our proposed methods ADHCG_1 and ADHCG_2 perform substantially better than the classical conjugate



(a) Performance based on iterations (b) Performance based on CPU time

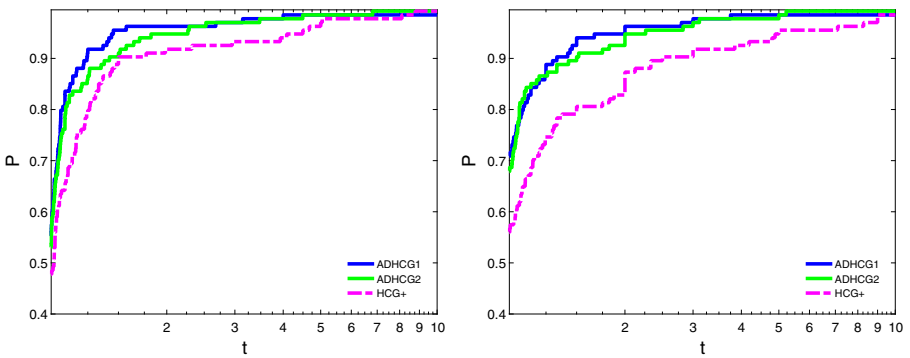
Fig. 2 Log_{10} scaled performance profiles for ADHCG_1, ADHCG_2, CG-DESCENT, and HYBRID based on number of iterations (**a**) and CPU time (**b**)



(a) Performance based on function evaluations (b) Performance based on gradient evaluations

Fig. 3 Log_{10} scaled performance profiles for ADHCG_1, ADHCG_2, and HCG+ based on number of function evaluations (**a**) and number of gradient evaluations (**b**)

gradient method HCG+. Figure 3 reports that ADHCG_1 illustrates the highest probability of being the most robust solver, followed by ADHCG_2, as regards the computational cost. In particular, ADHCG_1 solves about 53 and 56% of the test problems with the least function evaluations and gradient evaluations. Moreover, ADHCG_2 solves about 52.2% of the test problems while HCG+ solves only 43.2 and 43.2%, in the same situations. Furthermore, the interpretation of Fig. 4a shows that ADHCG_1 exhibits the best performance with respect to the number of iterations since it corresponds to the top curve. As regards the CPU time, Fig. 4b shows that our proposed methods exhibit the best performance, significantly outperforming HCG+. More analytically, ADHCG_1 and ADHCG_2 solve about 69.4 and 67.9% of the test problems, respectively with the least CPU time while HCG+ solves about 55.9% of the test problems. Since all conjugate gradient methods have been implemented with the same line search, we conclude that our proposed methods generate the best search directions on average.



(a) Performance based on iterations (b) Performance based on CPU time

Fig. 4 Log_{10} scaled performance profiles for ADHCG_1, ADHCG_2, and HCG+ based on number of iterations (**a**) and CPU time (**b**)

6 Conclusions

In this work, we presented a new conjugate gradient method incorporating the approach of the hybridization of the update parameters of DY and HS+ convexly in which the computation of the hybrization parameter is based on a quasi-Newton philosophy. More specifically, the value of the parameter is obtained by minimizing the distance between the hybrid conjugate gradient direction matrix and the self-scaling memoryless BFGS update. Moreover, an important property of our proposed method is that it ensures sufficient descent independent of the accuracy of the line search. Numerical comparisons have been made between our proposed method and the classical conjugate gradient methods CG-DESCENT [30], hybrid-enriched [40], and HCG+ [13] on a set of unconstrained optimization problems of the CUTEr collection. The reported numerical results demonstrated the computational efficiency and robustness of our proposed method.

In our future work, we intend to pursue an approach similar to [14, 17], studying the eigenvalues and the singular values of the update matrix. Since our numerical experiments are quite encouraging, another interesting aspect for future research is to perform a similar study as Zou et al. [62] and apply our proposed method on several synthetic function problems allowing control of the clustering of eigenvalues in the Hessian spectrum. Thus, we could examine the method's sensitivity to various degrees of ill conditioning and evaluate its computational performance as the condition number increases.

References

1. Al-Baali, M.: Descent property and global convergence of the Fletcher-Reeves method with inexact line search. IMA J. Numer. Anal. **5**, 121–124 (1985)
2. Al-Baali, M.: Analysis of a family self-scaling quasi-Newton methods. Comput. Optim. Appl. **9**, 191–203 (1998)
3. Al-Baali, M.: Numerical experience with a class of self-scaling quasi-Newton algorithms. J. Optim. Theory **96**, 533–553 (1998)
4. Al-Baali, M.: Extra updates for the BFGS method. Optim. Method Softw. **13**, 159–179 (2000)
5. Al-Baali, M., Spedicato, E., Maggioni, F.: Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems. Optim. Method Softw. **29**(5), 937–954 (2014)
6. Alekseev, A.K., Navon, I.M., Steward, J.L.: Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems. Optim. Method Softw. **24**(1), 63–87 (2009)
7. Andrei, N.: Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optim. Method Softw. **22**, 561–571 (2007)
8. Andrei, N.: Another hybrid conjugate gradient algorithm for unconstrained optimization. Numer. Algo. **47**, 143–156 (2008)
9. Andrei, N.: Hybrid conjugate gradient algorithm for unconstrained optimization. J. Optim. Theory Appl. **141**, 249–264 (2009)
10. Andrei, N.: Accelerated hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization. Numer. Algo. **54**, 23–46 (2010)
11. Apostolopoulou, M.S., Sotiropoulos, D.G., Livieris, I.E., Pintelas, P.: A Memoryless BFGS neural network training algorithm. In: 7Th IEEE International Conference on Industrial Informatics (INDIN'09), pp. 216–221 (2009)
12. Babaie-Kafaki, S., Fatemi, M., Mahdavi-Amiri, N.: Two effective hybrid conjugate gradient algorithms based on modified BFGS updates. Numer. Algo. **58**, 315–331 (2011)

13. Babaie-Kafaki, S., Ghanbari, R.: Two hybrid nonlinear conjugate gradient methods based on a modified secant equation. Optimization: A journal of mathematical programming and operations research, 1–16 (2012)
14. Babaie-Kafaki, S., Ghanbari, R.: The Dai-Liao nonlinear conjugate gradient method with optimal parameter choices. Eur. J. Oper. Res. **234**(3), 625–630 (2014)
15. Babaie-Kafaki, S., Ghanbari, R.: A hybridization of the Hestenes–Stiefel and Dai–Yuan conjugate gradient methods based on a least-squares approach. Optim. Method Softw. **30**(4), 673–681 (2015)
16. Babaie-Kafaki, S., Ghanbari, R.: A hybridization of the Polak-Ribière-Polyak and Fletcher-Reeves conjugate gradient methods. Numer. Algo. **68**(3), 481–495 (2015)
17. Babaie-Kafaki, S., Ghanbari, R.: Two optimal Dai-Laio conjugate gradient methods. Optimization **64**, 2277–2287 (2015)
18. Babaie-Kafaki, S., Ghanbari, R.: A class of adaptive Dai-Laio conjugate gradient methods based on scaled memoryless BFGS update 4OR, 1–8 (2016)
19. Babaie-Kafaki, S., Mahdavi-Amiri, N.: Two modified hybrid conjugate gradient methods based on a hybrid secant equation. Math. Model. Anal. **18**(1), 32–52 (2013)
20. Bongartz, I., Conn, A., Gould, N., Toint, P.: CUTE: Constrained and unconstrained testing environments. ACM Trans. Math. Softw. **21**(1), 123–160 (1995)
21. Burstedde, C., Kunoth, A.: The conjugate gradient method for linear ill-posed problems with operator perturbations. Numer. Algo. **48**(1), 161–188 (2008)
22. Dai, Y.H., Kou, C.X.: A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM J. Optim. **23**, 296–320 (2013)
23. Dai, Y.H., Yuan, Y.X.: A nonlinear conjugate gradient with a strong global convergence properties. SIAM J. Optim. **10**, 177–182 (1999)
24. Dai, Y.H., Yuan, Y.X.: Nonlinear Conjugate Gradient Methods. Shanghai Scientific and Technical Publishers, Shanghai (2000)
25. Dai, Y.H., Yuan, Y.X.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**, 33–47 (2001)
26. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
27. Fletcher, R. Practical Methods of Optimization, Volume 1: Unconstrained Optimization, 1st edition. Wiley, New York (1987)
28. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J **7**, 149–154 (1964)
29. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J Optim. **2**(1), 21–42 (1992)
30. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J Optim. **16**, 170–192 (2005)
31. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pacific J Optim. **2**, 35–58 (2006)
32. Hestenes, M.R., Stiefel, E.: Methods for conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
33. Hu, Y.F., Storey, C.: Global convergence result for conjugate gradient methods. J. Optim. Theory Appl. **71**, 399–405 (1991)
34. Kou, C.X., Dai, Y.H.: A modified self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno method for unconstrained optimization. Journal of Optimization Theory and Applications (2014)
35. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization methods. Math. Program. **45**, 503–528 (1989)
36. Liu, Q.: Two minimal positive bases based direct search conjugate gradient methods for computationally expensive functions. Numer. Algo. **58**(4), 461–474 (2011)
37. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, part 1: theory. J. Optim. Theory Appl. **69**, 129–137 (1991)
38. Livieris, I.E., Pintelas, P.: A new conjugate gradient algorithm for training neural networks based on a modified secant equation. Appl. Math. Comput. **221**, 491–502 (2013)
39. Livieris, I.E., Pintelas, P.: A limited memory descent Perry conjugate gradient method. Optim. Lett. **10**, 17–25 (2016)
40. Morales, J.L., Nocedal, J.: Enriched methods for large-scale unconstrained optimization. Comput. Optim. Appl. **21**, 143–154 (2002)

41. Nash, S.G.: Newton-type minimization via the Lanczos method. SIAM J Numer. Anal. **21**, 770–788 (1984)
42. Nash, S.G.: Preconditioning of truncated Newton methods. SIAM J Sci. Stat. Comput. **6**, 599–616 (1985)
43. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**(151), 773–782 (1980)
44. Nocedal, J.: Theory of algorithms for unconstrained optimization. Acta Numerica **1**, 199–242 (1992)
45. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
46. Nocedal, J., Yuan, Y.: Analysis of a self-scaling quasi-Newton method. Math. Program. **61**, 19–37 (1993)
47. Oren, S.S.: Self-Scaling Variable Metric Algorithms for Unconstrained Minimization. PhD Thesis, Stanford University, California (1972)
48. Oren, S.S., Luenberger, D.G.: Self-scaling variable metric (SSVM) algorithms, Part I: criteria and sufficient conditions for scaling a class of algorithms. Manag. Sci. **20**, 845–862 (1974)
49. Oren, S.S., Spedicato, E.: Optimal conditioning of self-scaling variable metric algorithms. Math. Program. **10**, 70–90 (1976)
50. Perry, J.M.: A Class of Conjugate Gradient Algorithms with a Two-Step Variable-Metric Memory. Center for Mathematical Studies in Economies and Management Science. Northwestern University Press, Evanston Illiois (1977)
51. Plato, R.: The conjugate gradient method for linear ill-posed problems with operator perturbations. Numer. Algo. **20**(1), 1–22 (1999)
52. Polak, E., Ribière, G.: Note sur la convergence de methods de directions conjuguees. Revue Francais d'Informatique et de Recherche Operationnelle **16**, 35–43 (1969)
53. Powell, M.J.D.: Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In: Cottle, R.W., Lemke, C.E. (eds.) Nonlinear Programming, SIAM-AMS Proceedings, vol. IX, pp. 53–72. SIAM Publications (1976)
54. Powell, M.J.D.: Restart procedures for the conjugate gradient method. Math. Program. **12**, 241–254 (1977)
55. Powell, M.J.D.: Nonconvex Minimization Calculations and the Conjugate Gradient Method. In: Numerical Analysis, Volume 1066 of Lecture Notes in Mathematics, pp. 122–141. Springer, Berlin (1984)
56. Risler, F., Rey, C.: Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. Numerical Algorithms, 23(1) (2000)
57. Shanno, D.F.: On the convergence of a new conjugate gradient algorithm. SIAM J. Numer. Anal. **15**(6), 1247–1257 (1978)
58. Touati-Ahmed, D., Storey, C.: Efficient hybrid conjugate gradient techniques. J. Optim. Theory Appl. **64**, 379–397 (1990)
59. Wu, X., Silva, B., Yuan, J.: Conjugate gradient method for rank deficient saddle point problems. Numer. Algo. **35**(2), 139–154 (2004)
60. Zhang, L., Zhou, W.: Two descent hybrid conjugate gradient methods for optimization. J. Comput. Appl. Math. **216**, 251–164 (2008)
61. Zhang, L., Zhou, W., Li, D.: Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search. Numer. Math. **104**, 561–572 (2006)
62. Zou, X., Navon, I.M., Berger, M., Phua, K.H., Schlick, T., Le Dimet, F.X.: Numerical experience with limited-memory quasi-Newton and truncated Newton methods. SIAM J Optim. **3**(3), 582–608 (1993)