mathematics

Article

A New Hybrid Descent Algorithm for Large-Scale Nonconvex Optimization and Application to Some Image Restoration Problems

Shuai Wang [1], Xiaoliang Wang [2,*], Yuzhu Tian [3] and Liping Pang [1]

[1] School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China; gacktshuaishuai@163.com (S.W.); lppang@dlut.edu.cn (L.P.)
[2] Department of Mathematics and Science, School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China
[3] School of Mathematics, Liaoning Normal University, Dalian 116026, China; tianyuzhu@lnnu.edu.cn
* Correspondence: xliangwang@126.com

Abstract: Conjugate gradient methods are widely used and attractive for large-scale unconstrained smooth optimization problems, with simple computation, low memory requirements, and interesting theoretical information on the features of curvature. Based on the strongly convergent property of the Dai–Yuan method and attractive numerical performance of the Hestenes–Stiefel method, a new hybrid descent conjugate gradient method is proposed in this paper. The proposed method satisfies the sufficient descent property independent of the accuracy of the line search strategies. Under the standard conditions, the trust region property and the global convergence are established, respectively. Numerical results of 61 problems with 9 large-scale dimensions and 46 ill-conditioned matrix problems reveal that the proposed method is more effective, robust, and reliable than the other methods. Additionally, the hybrid method also demonstrates reliable results for some image restoration problems.

Keywords: hybrid conjugate gradient method; acceleration scheme; sufficient descent property; global convergence; ill-conditioned matrix; image restoration

MSC: 65K05; 90C26

Citation: Wang, S.; Wang, X.; Tian, Y.; Pang, L. A New Hybrid Descent Algorithm for Large-Scale Nonconvex Optimization and Application to Some Image Restoration Problems. *Mathematics* **2024**, *12*, 3088. https://doi.org/10.3390/math12193088

Academic Editors: Moudafi Abdellatif and Andrea Scozzari

Received: 18 August 2024
Revised: 25 September 2024
Accepted: 27 September 2024
Published: 2 October 2024

1. Introduction

In this paper, we consider the following unconstrained problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, bound below, and its gradient is available. There are many effective methods for problem (1), such as Newton-type methods, quasi-Newton-type methods, spectral gradient methods, and conjugate gradient (CG for abbreviation) methods [1–11], etc. Meanwhile, there are also various free gradient optimization tools such as Nelder–Mead, generalized simulated annealing, and genetic algorithm [12–14], etc., for problem (1). In this part, we focus on CG methods and propose a new hybrid CG method for a large-scale problem (1). Actually, CG methods are one of the most effective methods for unconstrained problems, especially for large-scale cases, due to their low storage and globally convergent properties [3], in which the iterative point is usually generated by

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

where x_k is the current iteration; the scalar $\alpha_k > 0$ is the step length, determined by some line search strategy; and d_k is the search direction, defined by

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

where $g_k := g(x_k) = \nabla f(x_k)$ and β_k is called the conjugate parameter. A number of CG methods have been proposed by various modifications of the direction d_k and the parameter β_k; see [4–11,15–20], etc. Some CG methods have strong convergence properties, but their numerical performances may not be good in practice due to the jamming phenomenon [4]. These methods include Fletcher–Reeves (FR) [5], Dai–Yuan (DY) [6], and Fletcher (CD) [7], with the following conjugate parameters:

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

where $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$, and $\| \cdot \|$ stands for the Euclidean norm. On the other hand, some other CG methods may perform well in practice, but their convergence may be not guaranteed, especially for nonconvex functions. These methods include Hestenes–Stiefel (HS) [8], Polak–Ribière–Polyak (PRP) [9,10], and Liu–Storey (LS) [11], with the following conjugate parameters:

$$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \quad \beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_{k+1}^{LS} = -\frac{g_{k+1}^T y_k}{g_k^T d_k}.$$

In fact, these methods possess an automatically approximate restart feature which can avoid the jamming phenomenon, that is, when the step s_k is small, the factor y_k tends to zero, resulting in the conjugate parameter β_{k+1} becoming small and the new direction d_{k+1} approximating to the steepest descent direction $-g_{k+1}$.

To attain good computational performance and maintain the attractive feature of strong global convergence, many scholars have paid special attention to hybridizing these CG methods. Specifically, the authors in [21] proposed a hybrid PRP-FR CG method (H1 method in [22]) and the corresponding conjugate parameter was defined as $\beta_{k+1}^{H1} = \max\left\{0, \min\{\beta_{k+1}^{FR}, \beta_{k+1}^{PRP}\}\right\}$. Moreover, based on the above hybrid conjugate parameter, a new form was proposed in [23], where the parameter was defined by $\beta_{k+1} = \max\{-\beta_{k+1}^{FR}, \min\{\beta_{k+1}^{FR}, \beta_{k+1}^{PRP}\}\}$, and the global convergence property was established for the general function without the convexity assumption. In [24], a hybrid of the HS method and DY method was proposed in which the conjugate parameter was defined by $\beta_{k+1}^{H2} = \max\left\{0, \min\{\beta_{k+1}^{HS}, \beta_{k+1}^{DY}\}\right\}$. The numerical results indicated that the above hybrid method was more effective than the PRP algorithm. In the above hybrid CG methods, the search direction was in the form of (3). Moreover, the authors in [25] proposed a new hybrid three-term method in which the conjugate parameter is β_{k+1}^{H2} and the direction is $d_{k+1} = -g_{k+1} + (1 - \lambda_{k+1})\beta_{k+1}^{H2} d_k + \lambda_{k+1}\theta_{k+1} g_k$, where λ_{k+1} is the convex parameter. The above hybrid method demonstrates attractive numerical performance. Furthermore, in [22], the authors proposed two new hybrid methods based on the above conjugate parameters with different search directions. Concretely, the directions have the following common form:

$$d_{k+1} = -\left(1 + \beta_{k+1} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}\right) g_{k+1} + \beta_{k+1} d_k, \tag{4}$$

where $\beta_{k+1} = \beta_{k+1}^{H1}$ or $\beta_{k+1} = \beta_{k+1}^{H2}$. A remarkable feature of the above directions is that the sufficient descent property is automatically satisfied, independent of the accuracy of the line search strategy.

Motivated by the above discussions, in this paper, we propose a new hybrid descent CG method for large-scale nonconvex problems. The proposed hybrid method automatically enjoys the sufficient descent property independent of the accuracy of the line search technique. Furthermore, the global convergence for the general functions without convexity is established under the standard conditions. Numerical results of 549 large-scale problems and 46 ill-conditioned matrix problems indicate the proposed method is attractive and promising. Finally, we also apply the proposed method to some image restoration problems, which also verifies its reliability and effectiveness.

The rest of the paper is organized as follows. In Section 2, we propose a descent hybrid CG method which is based on the MHS method and DY method. Moreover, the sufficient descent property is satisfied independent of the accuracy of the line search techniques. Global convergence is established for the general function in Section 3. Numerical results are given in Section 4 to indicate the effectiveness and reliability of the proposed algorithm. Finally, some conclusions are presented.

2. Motivation, Algorithm, and Sufficient Descent Property

As mentioned in the above section, the HS method is generally regarded as one of the most effective CG methods, but its global convergence for general nonlinear functions is still erratic. Additionally, the HS method does not guarantee the descent property during the iterative process, that is, the condition $g_k^T d_k < 0$ may not be satisfied for $\forall\, k \geq 1$. Therefore, many researchers have been devoted to designing some descent HS conjugate gradient methods [4,24,26–30], etc. Specifically, to obtain an intuitively modified conjugate parameter, the authors in [26] approximated the direction d_{k+1}^{THS} by the two-term direction (3), where d_{k+1}^{THS} was defined by (4) with $\beta_{k+1} = \beta_{k+1}^{HS}$. Concretely, the least squares problem $\min_\beta \| -g_{k+1} + \beta_{k+1} d_k - d_{k+1}^{THS}\|^2$ was solved. After some algebraic manipulations, the unique solution was

$$\beta_{k+1}^{MHS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}\left(1 - \frac{(g_{k+1}^T d_k)^2}{\|g_{k+1}\|^2\|d_k\|^2}\right) = \beta_{k+1}^{HS}\vartheta_k, \tag{5}$$

where

$$\vartheta_k = 1 - \frac{(g_{k+1}^T d_k)^2}{\|g_{k+1}\|^2\|d_k\|^2}. \tag{6}$$

The above parameter and its modifications have some nice theoretical properties [26] and the method with (5) and (3) performs well. Meanwhile, it is clear that if the exact line search is adopted (i.e., $g_{k+1}^T d_k = 0$), it holds that $\beta_{k+1}^{MHS} = \beta_{k+1}^{HS} = \beta_{k+1}^{PRP}$.

To attain attractive computational performance and good theoretical properties, many researchers have proposed hybrid CG methods. Among these methods, hybridizations of the HS method and the DY method have shown promising numerical performance [31–34], etc. The HS method has a nice property of automatically satisfying the conjugate condition $d_{k+1}^T y_k = 0$ for $\forall\, k \geq 0$ independent of the accuracy of the line search strategies and the convexity of the objective function and performs well in practice. On the other hand, the DY method has remarkable convergence properties. These characteristics motivate us to propose new hybridizations of the HS method and the DY method which not only have attractive theoretical properties but also better numerical performance for large-scale nonconvex problems.

In the following, we focus on the conjugate parameter β_{k+1}^{MHS} and propose a new hybrid conjugate parameter of β_{k+1}^{DY} and β_{k+1}^{MHS}:

$$\beta_{k+1}^{N} = \max\left\{0, \min\left\{\beta_{k+1}^{DY}, \beta_{k+1}^{MHS}\right\}\right\}. \tag{7}$$

Now, based on the new hybrid conjugate parameter β_{k+1}^{N} and the modified descent direction (8), we propose our hybrid algorithm (NMHSDY) in detail.

It should be noted that the line search technique in Algorithm 1 is not fixed: It can be selected by the users. Next, we show that the search direction d_k generated by Algorithm 1 automatically has a sufficient descent property independent of any line search strategy.

Algorithm 1 New descent hybrid algorithm of MHS and DY methods (NMHSDY) for nonconvex functions.

Step 0 . Input and Initialization. Select an initial point $x_0 \in \mathbb{R}^n$, parameter $\varepsilon \geq 0$ and compute $f_0 = f(x_0)$ and $g_0 = g(x_0)$. Set $d_0 = -g_0$ and $k = 0$;
Step 1. If $\|g_k\| \leq \varepsilon$, then stop;
Step 2. Compute step length α_k along direction d_k by some line search strategy;
Step 3. Let $x_{k+1} = x_k + \alpha_k d_k$;
Step 4. Compute the conjugate parameter β_{k+1}^N by (7) and the search direction d_{k+1} by

$$d_{k+1} = -\left(1 + \beta_{k+1}^N \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}\right) g_{k+1} + \beta_{k+1}^N d_k, \tag{8}$$

Step 5. Set $k := k + 1$ and go to Step 2.

Theorem 1. *Let the search direction d_k be defined by (8) in Algorithm 1. Then, for any line search strategy, the sufficient descent property holds for nonconvex function $f(x)$, that is,*

$$g_k^T d_k = -\|g_k\|^2, \quad \forall\, k \geq 0. \tag{9}$$

Proof. By the definition of d_{k+1} in (8), we have

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \beta_{k+1} g_{k+1}^T d_k + \beta_{k+1} g_{k+1}^T d_k = -\|g_{k+1}\|^2.$$

Since $d_0 = -g_0$, then $d_0^T g_0 = -\|g_0\|^2$. All in all, (9) holds. This completes the proof. □

3. Convergence for General Nonlinear Functions

In this section, the global convergence of the NMHSDY method is presented. Before that, some common assumptions are listed.

Assumption 1. *The level set $\mathbb{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded, where x_0 is the initial point, i.e., there exists a positive constant $D > 0$ such that*

$$\|x\| \leq D, \quad \forall\, x \in \mathbb{L}. \tag{10}$$

Assumption 2. *In some neighborhood $\mathbb{N}$ of $\mathbb{L}$, the gradient $g(x) = \nabla f(x)$ is Lipschitz continuous, i.e., there exists a constant $L_1 > 0$ such that*

$$\|g(x) - g(y)\| \leq L_1 \|x - y\|, \quad \forall\, x, y \in \mathbb{N}. \tag{11}$$

Based on the above assumptions, we further obtain that there exists a constant $M > 0$ such that

$$\|g(x)\| \leq M, \quad \forall\, x \in \mathbb{L}. \tag{12}$$

In fact, it holds that $\|g(x)\| = \|g(x) - g(x_0) + g(x_0)\| \leq \|g(x) - g(x_0)\| + \|g(x_0)\| \leq L_1\|x - x_0\| + \|g(x_0)\| \leq 2L_1 D + \|g(x_0)\|$; hence, M can be $2L_1 E + \|g(x_0)\|$ or larger than that.

The line search strategy is another important element in iterative methods. In this part, we take the standard Wolfe line search strategy:

$$f(x_k + \alpha_k d_k) \leq f_k + \sigma_1 \alpha_k g_k^T d_k, \quad g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k, \tag{13}$$

where $0 < \sigma_1 < \sigma_2 < 1$. By property (9) and line search (13), it is satisfied that

$$f_{k+1} \leq f_k - \sigma_1 \alpha_k \|g_k\|^2 \leq f_k,$$

that is, the sequence $\{f_k\}$ is non-increasing and the sequence $\{x_k\}$ generated by Algorithm 1 is contained in the level set $\mathbb{L}$. Since f is continuously differentiable and the set $\mathbb{L}$ is bounded, then there exists a constant f^* such that

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

The Zoutendijk condition [35] plays an essential role in the global convergence of nonlinear CG methods. For completeness, we here state the lemma but omit its proof.

Lemma 1. *Suppose that Assumptions 1 and 2 hold. Consider any nonlinear CG method, in which α_k is obtained by the standard Wolfe line search (13) and d_k is a descent direction ($g_k^T d_k < 0$). Then, we have*

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{14}$$

Thereafter, the convergence property is presented in the following theorem for the general functions without convexity assumption.

Theorem 2. *Let Assumptions 1 and 2 hold and the sequence $\{x_k\}$ be generated by the NMHSDY algorithm. Set $l_{k+1} = \frac{\beta_{k+1}^N}{\beta_{k+1}^{DY}}$, and if $l_{k+1} \in [-\frac{1-\sigma_2}{1+\sigma_2}, 1-\sigma_2]$ holds, then we have*

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

Proof. We now prove (15) by contradiction and assume that there exists a constant $\mu > 0$ such that

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

Let γ_{k+1} be $1 + \beta_{k+1}^N \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}$, then the direction (8) can be rewritten as

$$d_{k+1} = -\gamma_{k+1} g_{k+1} + \beta_{k+1}^N d_k.$$

After some algebraic manipulation, we have

$$\|d_{k+1}\|^2 = (\beta_{k+1}^N)^2 \|d_k\|^2 - 2\gamma_{k+1} g_{k+1}^T d_{k+1} - \gamma_{k+1}^2 \|g_{k+1}\|^2.$$

Dividing both sides of the above equality by $(g_{k+1}^T d_{k+1})^2$, from (9), we have

$$\begin{aligned}
\frac{\|d_{k+1}\|^2}{\|g_{k+1}\|^4} &= (\beta_{k+1}^N)^2 \frac{\|d_k\|^2}{\|g_{k+1}\|^4} + \frac{2\gamma_{k+1}}{\|g_{k+1}\|^2} - \frac{\gamma_{k+1}^2}{\|g_{k+1}\|^2} \\
&= (\beta_{k+1}^N)^2 \frac{\|d_k\|^2}{\|g_{k+1}\|^4} + \frac{1}{\|g_{k+1}\|^2} - \frac{(\gamma_{k+1}-1)^2}{\|g_{k+1}\|^2}, \\
&= l_{k+1}^2 \frac{\|d_k\|^2}{(d_k^T y_k)^2} + \frac{1}{\|g_{k+1}\|^2} - \frac{(\gamma_{k+1}-1)^2}{\|g_{k+1}\|^2}, \\
&\leq \frac{l_{k+1}^2}{(1-\sigma_2)^2} \frac{\|d_k\|^2}{\|g_k\|^4} + \frac{1}{\|g_{k+1}\|^2} \leq \frac{\|d_k\|^2}{\|g_k\|^4} + \frac{1}{\|g_{k+1}\|^2},
\end{aligned} \tag{17}$$

where the first inequality holds by $d_k^T y_k \geq (\sigma_2 - 1) g_k^T d_k = (1 - \sigma_2)\|g_k\|^2$ and the last inequality holds by the bound for the scale l_{k+1}. By (17) and $\|d_0\|^2 = \|g_0\|^2$, it holds that

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq \sum_{i=0}^{k} \frac{1}{\|g_i\|^2}.$$

Then, by the above inequality and (16), it follows that

$$\frac{\|g_k\|^4}{\|d_k\|^2} \geq \frac{\mu^2}{k+1},$$

which indicates that

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} = \sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \infty,$$

which contradicts the Zoutendijk condition (14). So, (15) holds. □

Remark 1. *In [24], the authors presented a class of hybrid conjugate parameters, one of which is* $\beta_{k+1} = \max\{0, \min\{\beta_{k+1}^{DY}, \beta_{k+1}^{HS}\}\}$*, with the corresponding interval for* l_{k+1} *being* $[-(1-\sigma_2)/(1+\sigma_2),\ 1]$*. It is reasonable that the interval in our paper is smaller since we take* β_{k+1}^{MHS} *instead of* β_{k+1}^{HS} *and* $0 < \vartheta_k \leq 1$*.*

In the following, we discuss the global convergence of Algorithm 1 for general nonlinear functions in the case of $l_{k+1} \notin [-\frac{1-\sigma_2}{1+\sigma_2}, 1-\sigma_2]$. Motivated by the modified secant conditions in [36,37], in this part, based on the Wolfe line search strategy (13), we consider the following settings:

$$\bar{y}_k = y_k + m s_k, \tag{18}$$

where $m > 0$ is a constant. With the above setting, the modified conjugate parameter becomes

$$\beta_{k+1}^{NN} = \max\left\{0, \min\left\{\beta_{k+1}^{NDY}, \beta_{k+1}^{NMHS}\right\}\right\}, \tag{19}$$

where β_{k+1}^{NDY} and β_{k+1}^{NMHS} are, respectively,

$$\beta_{k+1}^{NDY} = \frac{\|g_{k+1}\|^2}{\bar{y}_k^T s_k}, \quad \beta_{k+1}^{NMHS} = \frac{g_{k+1}^T \bar{y}_k}{\bar{y}_k^T s_k}\left(1 - \frac{(g_{k+1}^T s_k)^2}{\|g_{k+1}\|^2 \|s_k\|^2}\right).$$

Meanwhile, the corresponding direction turns to

$$d_{k+1}^N = -\left(1 + \beta_{k+1}^{NN} \frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}\right) g_{k+1} + \beta_{k+1}^{NN} s_k, \tag{20}$$

The following lemma indicates the property of the scalar $\bar{y}_k^T s_k$ and $\|\bar{y}_k\|$.

Lemma 2. *Let* $\bar{y}_k$ *be defined by (18); then, adopting the Wolfe line search strategy (13), we obtain*

$$\bar{y}_k^T s_k \geq m\|s_k\|^2, \tag{21}$$

and

$$\|\bar{y}_k\| \leq (L_1 + m)\|s_k\|. \tag{22}$$

Proof. By the Wolfe line search strategy (13), we have

$$y_k^T s_k = (g_{k+1} - g_k)^T s_k \geq (\sigma_2 - 1) g_k^T s_k \geq (1 - \sigma_2)\alpha_k \|g_k\|^2 \geq 0,$$

which indicates $y_k^T s_k \geq 0$. Therefore, it holds that

$$\bar{y}_k^T s_k = y_k^T s_k + m\|s_k\|^2 \geq m\|s_k\|^2.$$

Hence, (21) holds. Meanwhile, we also have

$$\|\bar{y}_k\| = \|y_k + ms_k\| \leq \|y_k\| + m\|s_k\| \leq (L_1 + m)\|s_k\|,$$

where the second inequality holds by Assumption 2. Hence, (22) holds. This completes the proof. □

In the following, we assume that Algorithm 1 never stops and there exists a constant $\mu > 0$ such that for all k, (16) holds.

Lemma 3. *Suppose that Assumptions 1 and 2 and (16) hold. The sequences $\{x_k\}$ and $\{d_k^N\}$ are generated by Algorithm 1 with the conjugate parameter β_k^{NN} and adopting the Wolfe line search technique (13). Then, there exists a positive constant Γ such that*

$$\|g_k\| \leq \|d_k^N\| \leq (1 + 2\Gamma)\|g_k\|. \tag{23}$$

Proof. Based on the Wolfe line search technique (13), it holds that

$$\bar{y}_k^T d_k^N = y_k^T d_k^N + m\alpha_k\|d_k^N\|^2 \geq y_k^T d_k^N = -(1-\sigma_2)g_k^T d_k^N = (1-\sigma_2)\|g_k\|^2, \tag{24}$$

where the first inequality holds by the non-negativity of α_k and the last inequality holds by the sufficient descent property (9). Meanwhile, by (9) and the Cauchy–Schwartz inequality, it holds that, for $\forall\, k \geq 0$,

$$\|g_k\|\|d_k^N\| \geq -g_k^T d_k^N = \|g_k\|^2,$$

which implies that from condition (16),

$$\|d_k^N\| \geq \|g_k\| \geq \mu, \quad \forall\, k \geq 0. \tag{25}$$

By the definition of β_{k+1}^{NN}, we obtain that

$$\begin{aligned}
|\beta_{k+1}^{NN}| \quad &\leq \quad \max\left\{\frac{\|g_{k+1}\|^2}{|\bar{y}_k^T s_k|}, \frac{|g_{k+1}^T \bar{y}_k|}{|\bar{y}_k^T s_k|}\right\} \leq \max\left\{\frac{\|g_{k+1}\|^2}{m\|s_k\|^2}, \frac{\|g_{k+1}\|\|\bar{y}_k\|}{m\|s_k\|^2}\right\} \\
&\leq \quad \frac{\|g_{k+1}\|}{\|s_k\|}\max\left\{\frac{\|g_{k+1}\|}{m\|s_k\|}, \frac{L_1+m}{m}\right\} \leq \frac{\|g_{k+1}\|}{\|s_k\|}\max\left\{\frac{M}{m\bar{\alpha}\|d_k^N\|}, \frac{L_1+m}{m}\right\} \\
&\leq \quad \frac{\|g_{k+1}\|}{\|s_k\|}\max\left\{\frac{M}{m\bar{\alpha}\|g_k\|}, \frac{L_1+m}{m}\right\} \\
&\leq \quad \frac{\|g_{k+1}\|}{\|s_k\|}\max\left\{\frac{M}{m\bar{\alpha}\mu}, \frac{L_1+m}{m}\right\} := \Gamma\frac{\|g_{k+1}\|}{\|s_k\|},
\end{aligned}$$

where the second inequality holds by (21), the third inequality holds by (22), the fourth inequality holds by the condition $\alpha_k \geq \alpha > 0$ for all $k \geq 0$, the fifth inequality holds by (25), and the last inequality holds by the condition (16). Furthermore, we have

$$\left|\beta_{k+1}^{NN}\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}\right| \leq |\beta_{k+1}^{NN}|\frac{\|g_{k+1}\|\|s_k\|}{\|g_{k+1}\|^2} \leq \Gamma\frac{\|g_{k+1}\|}{\|s_k\|}\frac{\|s_k\|}{\|g_{k+1}\|} = \Gamma.$$

By the definition of d_k^N in (20) and the above discussions, it holds that

$$\begin{aligned}\|d_{k+1}^N\| &\leq \|g_{k+1}\| + \left|\beta_{k+1}^{NN}\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}\right|\|g_{k+1}\| + |\beta_{k+1}^{NN}|\|s_k\| \\ &\leq \|g_{k+1}\| + \Gamma\|g_{k+1}\| + \Gamma\frac{\|g_{k+1}\|}{\|s_k\|}\|s_k\| \\ &= (1+2\Gamma)\|g_{k+1}\|.\end{aligned}$$

With the help of (25), we conclude that

$$\|g_{k+1}\| \leq \|d_{k+1}^N\| \leq (1+2\Gamma)\|g_{k+1}\|.$$

Hence, (23) holds. This completes the proof. □

Theorem 3. *Suppose that Assumptions 1 and 2 hold. The sequences $\{x_k\}$ and $\{d_k^N\}$ are generated by Algorithm 1 with the conjugate parameter β_k^{NN} and the Wolfe line search technique (13) is adopted. Then, Algorithm 1 converges in the sense of (15).*

Proof. We prove the conclusion by contradiction and assume that there exists a positive constant μ such that (16) holds. Otherwise, Algorithm 1 converges in the sense of (15). From (9), we conclude that the new direction enjoys the sufficient descent property. Therefore, Lemma 1 holds, which implies that

$$+\infty = \sum_{k=1}^{+\infty}\frac{\mu^2}{(1+2\Gamma)^2} \leq \sum_{k=1}^{+\infty}\frac{\|g_k\|^2}{(1+2\Gamma)^2} = \sum_{k=1}^{+\infty}\frac{\|g_k\|^4}{(1+2\Gamma)^2\|g_k\|^2} \leq \sum_{k=1}^{+\infty}\frac{(g_k^T d_k^N)^2}{\|d_k^N\|^2} < +\infty,$$

where the first inequality holds by (16), the second inequality holds by (9) and (23), and the last inequality holds by Lemma 1. However, that is a contradiction and the assumption does not hold. So, the $\liminf_{k\to+\infty}\|g_k\| = 0$ holds. This completes the proof. □

4. Numerical Performance

In this section, we focus on the numerical performance of Algorithm 1 and compare it with several effective CG methods. In the experiment, we code these algorithms in Matlab 2016b and perform them on a PC computer, whose processor has AMD 2.10 GHz, RAM of 16.00 GB and the Windows 10 operating system.

4.1. Performance on Benchmark Problems

In this subsection, we check the performance of the NMHSDY method and compare it with two effective modified HS methods in [26,28] and the hybrid method in [24]. In [26], the authors proposed an effective modified HS method (MHSCG method for abbreviation) in which the conjugate parameter is

$$\beta_{k+1} = \max\{0, \bar{\beta}_{k+1}^{MHS}\}, \quad \bar{\beta}_{k+1}^{MHS} = \beta_{k+1}^{MHS} - \lambda\left(\frac{\|y_k\|\vartheta_k}{y_k^T d_k}\right)^2 g_{k+1}^T d_k,$$

where $\lambda > 1/4$ is a parameter. The direction in [26] is in the form of (3) and the corresponding method has attractive numerical performance. Dai and Kou in [28] introduced another effective class of CG schemes (DK+ method for abbreviation) depending on the parameter τ_k, where the corresponding conjugate parameter β_{k+1} is defined by

$$\beta_{k+1}^{DK}(\tau_k) = \beta_{k+1}^{HS} - \left(\tau_k + \frac{\|y_k\|^2}{y_k^T s_k} - \frac{y_k^T s_k}{\|s_k\|^2}\right)\frac{g_{k+1}^T s_k}{y_k^T d_k}, \quad s_k = x_{k+1} - x_k.$$

The direction in [28] is also in the form of (3). To establish global convergence for general nonlinear functions, a truncated strategy is used, that is,

$$\beta_{k+1}^{DK+}(\tau_k) = \max\left\{\beta_{k+1}^{DK}(\tau_k), \eta\frac{g_{k+1}^T d_k}{\|d_k\|^2}\right\},$$

where $\eta \in [0,1)$ is a parameter. The numerical results indicated the DK+ method has good and reliable numerical performance. Dai and Yuan, in [24], proposed an effective hybrid CG method (HSDY method for abbreviation) in which the conjugate parameter is

$$\beta_{k+1} = \max\{0, \min\{\beta_{k+1}^{DY}, \beta_{k+1}^{HS}\}\}.$$

The hybrid method also has global convergence and attractive numerical performance.

In the following, we focus on the numerical performance and the large-scale unconstrained problems in Table 1 (see [38] for details). In order to improve numerical performance, Andrei, in [39], proposed an accelerated strategy which modified the step in a multiplicative manner. In this part, we also utilize this strategy and regard Algorithm 1 with the accelerated strategy as Algorithm 1. To compare the conjugate parameters and the search directions fairly, here we adopt the Wolfe line search technique (13) for all methods.

Table 1. The test problems.

No.	Problem	No.	Problem
1	Extended Freudenstein and Roth Function	32	ARWHEAD (CUTE)
2	Extended Trigonometric Function	33	NONDIA (Shanno-78) (CUTE)
3	Extended Rosenbrock Function	34	DQDRTIC (CUTE)
4	Extended Beale Function	35	EG2 (CUTE)
5	Extended Penalty Function	36	DIXMAANA (CUTE)
6	Perturbed Quadratic Function	37	DIXMAANB (CUTE)
7	Raydan 1 Function	38	DIXMAANC (CUTE)
8	Raydan 2 Function	39	DIXMAANE (CUTE)
9	Diagonal 3 Function	40	Broyden Tridiagonal
10	Generalized Tridiagonal-1 Function	41	Almost Perturbed Quadratic
11	Extended Tridiagonal-1 Function	42	Tridiagonal Perturbed Quadratic
12	Extended Three Exponential Terms	43	EDENSCH Function (CUTE)
13	Generalized Tridiagonal-2	44	VARDIM Function (CUTE)
14	Diagonal 4 Function	45	LIARWHD (CUTE)
15	Diagonal 5 Function	46	DIAGONAL 6
16	Extended Himmelblau Function	47	DIXMAANF (CUTE)
17	Generalized PSC1 Function	48	DIXMAANG (CUTE)
18	Extended PSC1 Function	49	DIXMAANH (CUTE)
19	Extended Powell Function	50	DIXMAANI (CUTE)
20	Extended Cliff Function	51	DIXMAANJ (CUTE)
21	Quadratic Diagonal Perturbed Function	52	DIXMAANK (CUTE)
22	Extended Wood Function	53	DIXMAANL (CUTE)
23	Extended Hiebert Function	54	DIXMAAND (CUTE)
24	Quadratic Function QF1	55	ENGVAL1 (CUTE)
25	Extended Quadratic Penalty QP1 Function	56	COSINE (CUTE)
26	Extended Quadratic Penalty QP2 Function	57	Extended DENSCHNB (CUTE)
27	A Quadratic Function QF2 Function	58	Extended DENSCHNF (CUTE)
28	Extended EP1 Function	59	SINQUAD (CUTE)
29	Extended Tridiagonal-2 Function	60	Scaled Quadratic SQ1
30	BDQRTIC (CUTE)	61	Scaled Quadratic SQ2
31	TRIDIA (CUTE)		

In the experiment, for each problem we consider nine large-scale dimensions with 300, 600, 900, 3000, 6000, 9000, 30,000, 60,000 and 90,000 variables. The parameters used in the Wolfe line search are $\sigma_1 = 0.20$ and $\sigma_2 = 0.85$. The other parameters for the MHSCG method and the DK+ method are as default.

During the progress, the Himmeblau stopping rule is adopted: if $|f(x_k)| > \varepsilon_1$, let $stop1 = \frac{|f(x_k)-f(x_{k+1})|}{|f(x_k)|}$, otherwise, $stop1 = |f(x_k) - f(x_{k-1})|$. If the conditions $\|g_k\| \leq \varepsilon$ or $stop1 \leq \varepsilon_2$ are satisfied, then the progress is stopped, where the values of parameters ε, ε_1, and ε_2 are $\varepsilon = 10^{-6}$, and $\varepsilon_1 = \varepsilon_2 = 10^{-5}$. Meanwhile, we also stop the algorithm when the number of iterations is greater than 5000.

In order to present the performances of methods more intuitively, the tool in [40] is adopted to analyze the profiles of these methods. Robustness and efficiency rates are readable on the right and left vertical axes of the corresponding performance profiles, respectively. To present a detailed numerical comparison, two different scales have been considered for the τ-axis. One is $\tau \in [1, 1.5]$, which shows what happens for the values of τ near to 1. The other is used to present the trend for large values of τ. In Figures 1–3, we, respectively, show the performance of these methods relative to the number of iterations (NI), the number of function-gradient valuations (NFG; which is the sum of the number of function valuations and gradient valuations), and the CPU time consumed in seconds.

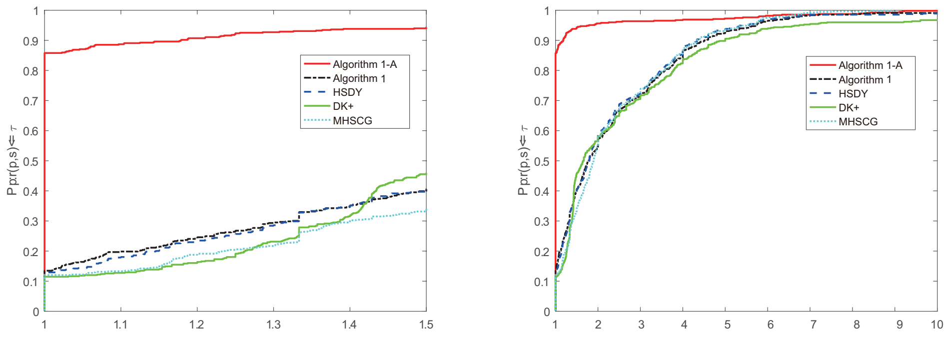

Figure 1. Performance profiles of the methods in the number of iterations case.

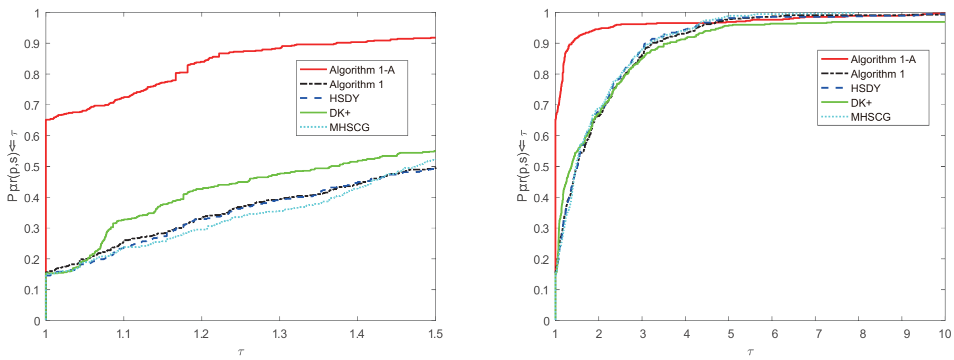

Figure 2. Performance profiles of the methods in the function and gradient case.

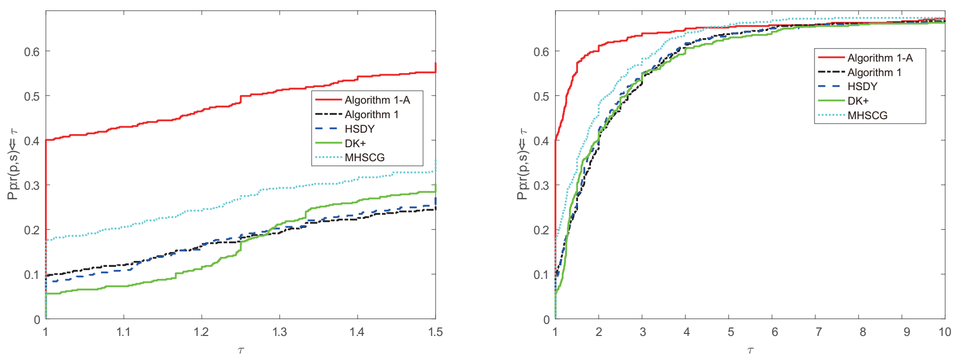

Figure 3. Performance profiles of the methods in the CPU time consumed case.

From Figures 1–3, we have that Algorithm 1 is comparable and a little more effective than the HSDY method, the DK+ method, and the MHSCG method for the above problems.

Meanwhile, Algorithm 1, with the accelerated strategy, is much effective and performs best in the experiment, which indicates that the accelerated technique indeed works and reduces the number of iterations and the number of function and gradient evaluations.

4.2. Comparison for Stability

In this subsection, we consider the numerical stability of Algorithm 1 for the ill-conditioned matrix problems and compare it with the MHSCG method in [26]. In fact, the quadratic objective function $f(x) = x^T \mathcal{I} x$ of (1) is ill-conditioned if matrix $\mathcal{I} \in \mathbb{R}^{n \times n}$ is in the form

$$\mathcal{I}_{i,j} = \frac{1}{i+j-1}, \quad i,j = 1,2,\ldots,n.$$

It is clear that the matrix $\mathcal{I}$ is ill-conditioned and positive definite [41], with different dimensions $n = 5, 6, \ldots, 50$. Furthermore, the authors in [42] show that the ℓ_∞ norm condition number of the Hessian matrix $\mathcal{I}$ gradually increases from 9.4366×10^5 for $n = 5$ to 6.9007×10^{20} for $n = 50$. In the following, we explore the numerical performance. The experimental environment, the parameter values, and the stop rule remain the same as in the above subsection. Meanwhile, the initial point is selected as $x_0 = (10, 10, \cdots, 10)$. The corresponding numerical results are presented in Tables 2 and 3, in which Dim is the dimension of x, NI means the number of iterations, NFG is the sum of the number of function and gradient evaluations, $Time$ means the CPU time consumed in seconds, and f^* denotes the optimal function obtained by the methods.

Table 2. Numerical results of the MHSCG method and Algorithm 1 in 5–40 dimensions.

	MHSCG Method				Algorithm 1			
Dim	**NI**	**NFG**	**Time**	f^*	**NI**	**NFG**	**Time**	f^*
5	5	22	0.000000	0.000000	5	22	0.000000	0.000000
6	13	48	0.031250	0.000000	13	48	0.000000	0.000000
7	21	72	0.000000	0.000000	21	72	0.000000	0.000000
8	21	72	0.000000	0.000000	21	72	0.015625	0.000000
9	27	92	0.000000	0.000006	24	82	0.000000	0.000000
10	41	143	0.000000	0.000007	29	97	0.000000	0.000000
11	75	246	0.000000	0.000009	55	192	0.000000	0.000000
12	58	203	0.000000	0.000003	70	249	0.000000	0.000008
13	57	190	0.000000	0.000004	70	246	0.000000	0.000000
14	86	276	0.031250	0.000008	123	429	0.000000	0.000000
15	75	264	0.031250	0.000009	89	314	0.000000	0.000009
16	98	330	0.000000	0.000008	52	177	0.000000	0.000001
17	24	82	0.000000	0.000002	66	232	0.000000	0.000002
18	65	215	0.031250	0.000009	28	97	0.000000	0.000000
19	31	105	0.000000	0.000007	27	94	0.000000	0.000000
20	111	373	0.031250	0.000009	131	452	0.000000	0.000010
21	117	379	0.031250	0.000008	31	108	0.000000	0.000000
22	96	333	0.000000	0.000010	66	236	0.000000	0.000000
23	68	233	0.000000	0.000009	85	303	0.031250	0.000000

Table 2. *Cont.*

	MHSCG Method				Algorithm 1			
Dim	**NI**	**NFG**	**Time**	f^*	**NI**	**NFG**	**Time**	f^*
24	50	175	0.031250	0.000000	43	151	0.000000	0.000000
25	96	318	0.000000	0.000009	103	377	0.000000	0.000000
26	113	370	0.000000	0.000009	112	410	0.000000	0.000004
27	22	78	0.000000	0.000000	129	473	0.031250	0.000001
28	104	372	0.000000	0.000009	24	87	0.000000	0.000000
29	156	513	0.000000	0.000010	130	449	0.000000	0.000001
30	54	185	0.000000	0.000001	121	446	0.000000	0.000007
31	39	144	0.000000	0.000003	97	341	0.000000	0.000008
32	26	97	0.000000	0.000001	79	292	0.031250	0.000001
33	87	309	0.000000	0.000010	66	234	0.031250	0.000002
34	58	213	0.000000	0.000006	60	217	0.031250	0.000000
35	97	327	0.000000	0.000009	25	95	0.000000	0.000002
36	24	86	0.000000	0.000000	132	481	0.000000	0.000003
37	23	84	0.000000	0.000004	55	192	0.000000	0.000001
38	93	312	0.046875	0.000010	40	142	0.000000	0.000003
39	148	487	0.046875	0.000005	124	442	0.140625	0.000010
40	66	231	0.000000	0.000007	136	496	0.031250	0.000007

Table 3. Numerical results of the MHSCG method and Algorithm 1 in 41–50 dimensions.

	MHSCG Method				Algorithm 1			
Dim	**NI**	**NFG**	**Time**	f^*	**NI**	**NFG**	**Time**	f^*
41	76	267	0.031250	0.000005	65	239	0.000000	0.000005
42	130	438	0.046875	0.000008	85	302	0.000000	0.000010
43	164	546	0.031250	0.000009	136	488	0.000000	0.000009
44	184	594	0.031250	0.000007	29	103	0.000000	0.000006
45	17	64	0.000000	0.000000	17	64	0.000000	0.000000
46	155	519	0.031250	0.000010	62	222	0.000000	0.000000
47	123	421	0.031250	0.000010	107	379	0.031250	0.000010
48	20	74	0.000000	0.000000	68	250	0.015625	0.000010
49	135	473	0.031250	0.000010	102	375	0.031250	0.000010
50	178	594	0.062500	0.000010	151	542	0.031250	0.000009

From Tables 2 and 3, it can be found that for the dimensions from 5 to 50, Algorithm 1 and the MHSCG method successfully solve all of them and obtain reasonable optimal function values, which are all not greater than 10^{-5}. For most problems, Algorithm 1 needed fewer iterations and function and gradient evaluations and obtained better optimal values. In order to show numerical performance intuitively, here we also adopt the performance profiles in [40] for the NI and NFG cases. The corresponding performance profiles are given in Figures 4 and 5.

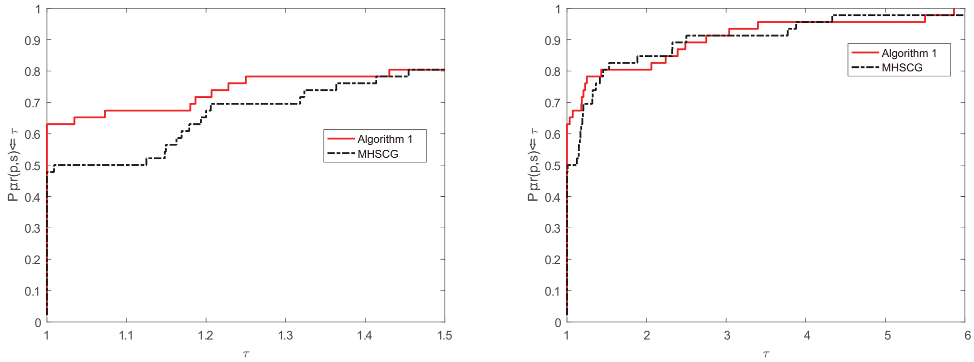

Figure 4. Performance profiles of Algorithm 1 and the MHSCG method in NI case.

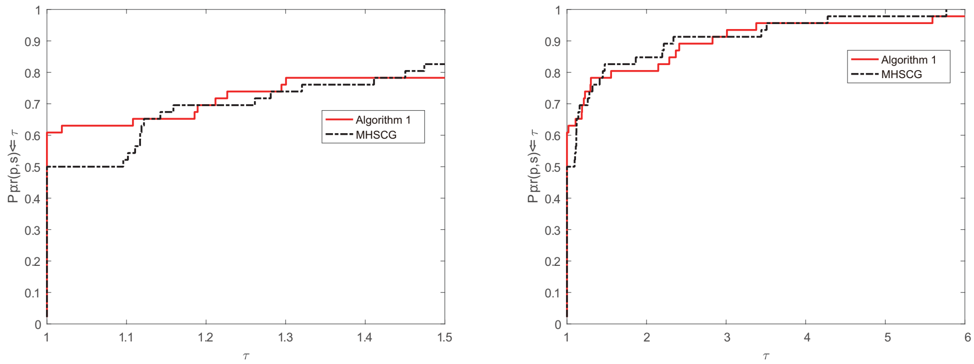

Figure 5. Performance profiles of Algorithm 1 and the MHSCG method in NFG case.

Figure 4 shows that Algorithm 1 and the MHSCG method solve these testing problems with the least total number of iterations in 63% and 48% of cases, respectively. Figure 5 indicates that Algorithm 1 and the MHSCG method solve these testing problems with the least total number of function and gradient evaluations in 61% and 50% of cases, respectively. All in all, the numerical results show that Algorithm 1 is more effective and stable than the MHSCG method for these ill-conditioned matrix problems.

4.3. Application to Image Restoration

In this subsection, we apply Algorithm 1 to some image restoration problems [43–45]. During the process, the normal Wolfe line search technique is adopted, the corresponding parameters remain unchanged, and two noise level cases for the Barbara.png (512 $\times$ 512) and Baboon.bmp (512 $\times$ 512) images are considered. In this part, we stop the process when the following criteria are both satisfied:

$$\frac{|f(x_{k+1}) - f(x_k)|}{|f(x_k)|} < 10^{-3}, \quad \|g(x_k)\| < 10^{-3}(1 + |f(x_k)|).$$

Meanwhile, to assess the restoration performance qualitatively, we also utilize the peak signal to noise ratio [45] (PSNR), which is defined by $PSNR = 10\log_{10} \frac{255^2}{\frac{1}{M\times N}\sum_{i,j}(u^r_{i,j} - u^*_{i,j})^2}$, where M and N are the true image pixels, and $u^r_{i,j}$ and $u^*_{i,j}$ denote the pixel values of the restored image and the original image, respectively. For the noise level, we consider two cases: 20% (a low-level case) and 60% (a high-level case). The consumed CPU time and the corresponding PSNR values are given in Table 4. Meanwhile, the detailed performances are presented in Figures 6 and 7, respectively.

Table 4. Test results for Algorithm 1 and the MHSCG method.

Image	Noise Level	Algorithm 1 PSNR	Algorithm 1 CPU Time	MHSCG Method PSNR	MHSCG Method CPU Time
Barbara	20%	29.6638	2.265625	29.5831	2.578125
Baboon	20%	27.9223	2.609375	27.8455	3.250000
Barbara	60%	23.1256	3.593750	23.1103	3.765625
Baboon	60%	21.1836	3.484375	21.1582	3.671875

Figure 6. The noisy Barbara image, corrupted by salt-and pepper noise (**the first column**); the images restored via Algorithm 1 (**the second column**), and via the MHSCG method (**the third column**).

Figure 7. The noisy Baboon image, corrupted by salt-and pepper noise (**the first column**); the images restored via Algorithm 1 (**the second column**), and via the MHSCG method **(the third column**).

From Table 4 and Figures 6 and 7, we have that Algorithm 1 and the MHSCG method can all solve the image restoration problems successfully within a suitable time, and Algorithm 1 seems to perform a little better than the MHSCG method.

5. Conclusions

Conjugate gradient methods are attractive and effective for large-scale unconstrained optimization smooth problems due to their simple computation and low memory requirements. The Dai–Yuan conjugate gradient method has good theoretical properties and generates a descent direction in each iteration. Whereas, the Hestenes–Stiefel conjugate gradient method automatically satisfies the conjugate condition $y_k^T d_{k+1} = 0$ without any line search technique and performs well in practice. By the above discussions, we propose a new descent hybrid conjugate gradient method. The proposed method has a sufficient descent property independent of any line search technique. Under some mild conditions, the proposed method is globally convergent. In the experiments, we first consider 61 unconstrained problems with 9 different dimensions up to 90,000. Thereafter, 46 ill-conditioned matrix problems are also tested. The primary numerical results show that the proposed method is more effective and stable. Finally, we apply the hybrid method to some image restoration problems. The results indicate our method is attractive and reliable.

Author Contributions: Conceptualization, S.W., X.W., Y.T. and L.P.; methodology, S.W. and X.W.; software, X.W.; validation, X.W., L.P. and Y.T.; formal analysis, X.W., Y.T. and L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Science Foundation of Zhejiang Sci-Tech University (ZSTU) under Grant No. 21062347-Y.

Data Availability Statement: All data included in this study are available upon reasonable request.

Conflicts of Interest: The authors declare no competing interests.

References

1. Li, D.; Fukushima, M. A global and superlinear convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **1999**, *37*, 152–172. [CrossRef]
2. Yuan, G.; Wei, Z.; Wang, Z. Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minimization. *Comput. Optim. Appl.* **2013**, *54*, 45–64. [CrossRef]
3. Dai, Y.; Han, J.; Liu, G.; Sun, D.; Yin, H.; Yuan, Y. Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.* **2000**, *10*, 345–358. [CrossRef]
4. Hager, W.; Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **2005**, *16*, 170–192. [CrossRef]
5. Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149–154. [CrossRef]
6. Dai, Y.; Yuan, Y. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **1999**, *10*, 177–182. [CrossRef]
7. Fletcher, R. *Practical Methods of Optimization*; Unconstrained Optimization; John Wiley & Sons: New York, NY, UAS, 1987; Volume 1.
8. Hestenes, R.; Stiefel, L. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [CrossRef]
9. Polyak, B.T. The conjugate gradient method in extreme problems. *USSR Comput. Math. Math. Phys.* **1969**, *9*, 94–112. [CrossRef]
10. Polak, E.; Ribière, G. Note sur la convergence de méthodes de directions conjuguées. *Rev. Fr. Informat Rech. Opér.* **1969**, *16*, 35–43. [CrossRef]
11. Liu, Y.; Storey, C. Efficient generalized conjugate gradient algorithms Part 1: Theory. *J. Optim. Theory Appl.* **1991**, *69*, 129–137. [CrossRef]
12. Xiang, Y.; Gong, X.G. Efficiency of generalized simulated annealing. *Phys. Rev. E* **2000**, *62*, 4473–4476. [CrossRef] [PubMed]
13. Yuan, Q.; Qian, F. A hybrid genetic algorithm for twice continuously differentiable NLP problems. *Comput. Chem. Eng.* **2010**, *34*, 36–41. [CrossRef]
14. Gao, F.C.; Han, L.X. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.* **2012**, *51*, 259–277. [CrossRef]
15. Yuan, G.; Wang, X.; Sheng, Z. The projection technique for two open problems of unconstrained optimization problems. *J. Optim. Theory Appl.* **2020**, *186*, 590–619. [CrossRef]
16. Yuan, G.; Wang, X.; Sheng, Z. Family weak conjugate gradient algorithms and their convergence analysis for nonconvex functions. *Numer. Algorithms* **2020**, *84*, 935–956. [CrossRef]
17. Mousavi, A.; Esmaeilpour, M.; Sheikhahmadi, A. A new family of Polak-Ribière-Polyak conjugate gradient method for impulse noise removal. *Soft Comput.* **2023**, *27*, 17515–17524. [CrossRef]
18. Polyak, B.T. *Introduction to Optimization*; Optimization Software Inc., Publications Division: New York, NY, USA, 1987.
19. Wang, X.; Yuan, G.; Pang, L. A class of new three-term descent conjugate gradient algorithms for large-scale unconstrained optimization and applications to image restoration problems. *Numer. Algorithms* **2023**, *93*, 949–970. [CrossRef]
20. Wang, X. A class of spectral three-term descent Hestenes-Stiefel conjugate gradient algorithms for large-scale unconstrained optimization and image restoration problems. *Appl. Numer. Math.* **2023**, *192*, 41–56. [CrossRef]

21. Touati-Ahmed, D.; Storey, C. Efficient hybrid conjugate gradient techniques. *J. Optim.Theory Appl.* **1990**, *64*, 379–397. [CrossRef]
22. Zhang, L.; Zhou, W. Two descent hybrid conjugate gradient methods for optimization. *J. Comput. Appl. Math.* **2008**, *216*, 251–264. [CrossRef]
23. Gilbert, J.; Nocedal, J. Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* **1992**, *2*, 21–42. [CrossRef]
24. Dai, Y.; Yuan, Y. An efficient hybrid conjugate gradient method for unconstrained optimization. *Ann. Oper. Res.* **2001**, *103*, 33–47. [CrossRef]
25. Jiang, X.; Liao, W.; Yin, J.; Jian, J. A new family of hybrid three-term conjugate gradient methods with applications in image restoration. *Numer. Algorithms* **2022**, *91*, 161–191. [CrossRef]
26. Amini, K.; Faramarzi, P.; Pirfalah, N. A modified Hestenes-Stiefel conjugate gradient method with an optimal property. *Optim. Methods Softw.* **2019**, *34*, 770–782. [CrossRef]
27. Narushima, Y.; Yabe, H.; Ford, J. A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. *SIAM J. Optim.* **2011**, *21*, 212–230. [CrossRef]
28. Dai, Y.; Kou, C. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **2013**, *23*, 296–320. [CrossRef]
29. Woldu, T.; Zhang, H.; Zhang, X.; Yemane, H. A modified nonlinear conjugate gradient algorithm for large-scale nonsmooth convex optimization. *J. Optim. Theory Appl.* **2020**, *185*, 223–238. [CrossRef]
30. 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.* **2016**, *168*, 129–152. [CrossRef]
31. Babaie-Kafaki, S.; Fatemi, M.; Mahdavi-Amiri, N. Two effective hybrid conjugate gradient algorithms based on modified BFGS updates. *Numer. Algorithms* **2011**, *58*, 315–331. [CrossRef]
32. Livieris, I.; Tampakas, V.; Pintelas, P. A descent hybrid conjugate gradient method based on the memoryless BFGS update. *Numer. Algorithms* **2018**, *79*, 1169–1185. [CrossRef]
33. Khoshgam, Z.; Ashrafi, A. A new hybrid conjugate gradient method for large-scale unconstrained optimization problem with non-convex objective function. *Comp. Appl. Math.* **2019**, *38*, 186. [CrossRef]
34. Narayanan, S.; Kaelo, P. A linear hybridization of Dai-Yuan and Hestenes-Stiefel conjugate gradient method for unconstrained optimization. *Numer.-Math.-Theory Methods Appl.* **2021**, *14*, 527–539.
35. Zoutendijk, G. *Nonlinear Programming, Computational Methods*; Integer & Nonlinear Programming: Amsterdam, The Netherlands, 1970; pp. 37–86.
36. Li, D.; Fukushima, M. A modified BFGS method and its global convergence in nonconvex minimization. *J. Comput. Appl. Math.* **2001**, *129*, 15–35. [CrossRef]
37. Babaie-Kafaki, S.; Ghanbari, R. A modified scaled conjugate gradient method with global convergence for nonconvex functions. B *Bull. Belg. Math. Soc. Simon Stevin* **2014**, *21*, 465–477. [CrossRef]
38. Andrei, N. An unconstrained optimization test functions collection. *Environ. Ence Technol.* **2008**, *10*, 6552–6558.
39. Andrei, N. An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. *Numer. Algorithms* **2006**, *42*, 63–73. [CrossRef]
40. Dolan, E.; Moré, J. Benchmarking optimization software with performance profiles. *Math. Program* **2002**, *91*, 201–213. [CrossRef]
41. Watkins, S. *Fundamentals of Matrix Computations*; John Wiley and Sons: New York, NY, USA, 2002.
42. Babaie-Kafaki, S. A hybrid scaling parameter for the scaled memoryless BFGS method based on the ℓ_∞ matrix norm. *Int. J. Comput. Math.* **2019**, *96*, 1595–1602. [CrossRef]
43. Yu, G.; Huang, J.; Zhou, Y. A descent spectral conjugate gradient method for impulse noise removal. *Appl. Math. Lett.* **2010**, *23*, 555–560. [CrossRef]
44. Yuan, G.; Lu, J.; Wang, Z. The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems. *Appl. Numer. Math.* **2020**, *152*, 1–11. [CrossRef]
45. Bovik, A. *Handbook of Image and Video Processing*; Academic: New York, NY, USA, 2000.