

Article

# A Dynamically Adjusted Subspace Gradient Method and Its Application in Image Restoration

Jun Huo <sup>1</sup>, Yuping Wu <sup>2</sup>, Guoen Xia <sup>3,†</sup> and Shengwei Yao <sup>1,4,\*,†</sup>

<sup>1</sup> Guangxi (ASEAN) Financial Research Center, Guangxi University of Finance and Economics, Nanning 530003, China; 2018210016@gxufe.edu.cn

<sup>2</sup> College of Mathematics and Information Science, Guangxi University, Nanning 530005, China; 1806301033@st.gxu.edu.cn

<sup>3</sup> School of Business Administration, Guangxi University of Finance and Economics, Nanning 530003, China; 2007110010@gxufe.edu.cn

<sup>4</sup> Guangxi Key Laboratory Cultivation Base of Cross-Border E-Commerce Intelligent Information Processing, Guangxi University of Finance and Economics, Nanning 530003, China

\* Correspondence: ysw@gxu.edu.cn

† These authors contributed equally to this work.

**Abstract:** In this paper, a new subspace gradient method is proposed in which the search direction is determined by solving an approximate quadratic model in which a simple symmetric matrix is used to estimate the Hessian matrix in a three-dimensional subspace. The obtained algorithm has the ability to automatically adjust the search direction according to the feedback from experiments. Under some mild assumptions, we use the generalized line search with non-monotonicity to obtain remarkable results, which not only establishes the global convergence of the algorithm for general functions, but also R-linear convergence for uniformly convex functions is further proved. The numerical performance for both the traditional test functions and image restoration problems show that the proposed algorithm is efficient.

**Keywords:** subspace method; conjugate gradient method; quadratic approximation; image restoration



**Citation:** Huo, J.; Wu, Y.; Xia, G.; Yao, S. A Dynamically Adjusted Subspace Gradient Method and Its Application in Image Restoration. *Symmetry* **2021**, *13*, 2450. <https://doi.org/10.3390/sym13122450>

Academic Editors: Liang-Jian Deng, Xian-Ming Gu and Tai-Xiang Jiang

Received: 8 December 2021

Accepted: 15 December 2021

Published: 20 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Conjugate Gradient (CG) method is dedicated to solving the unconstrained optimization problem:

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

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is smooth and the gradient of  $f(x)$  at  $x_k$  is marked  $g_k$ . The advantages of its simple form and low storage requirements make the CG method a powerful tool for dealing with problem (1). It starts at a starting point  $x_0$  and generates an iterative sequence  $\{x_k\}$  in the following form:

$$x_{k+1} = x_k + \alpha_k d_k, k \geq 0, \quad (2)$$

that is,  $x_k$  moves forward by one step  $\alpha_k$  along the search direction  $d_k$  and reaches the  $(k + 1)$ -th iteration point  $x_{k+1}$ .

The direction  $d_k$  is usually defined as

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

where  $\beta_k$  is CG parameter. The different  $\beta_k$  corresponds to different CG methods, such as Polak and Ribiere (PRP) [1], Hestenes and Stiefel (HS) [2], Liu and Storey (LS) [3], Fletcher and Reeves (FR) [4], Dai and Yuan (DY) [5], and the conjugate descent (CD) method [6]. In addition, more relevant research and the progress of CG method can be found in the literature [7–10].

The step size  $\alpha_k$  can be obtained by different rules. In this paper, we focus on the following generalized line search, which has been shown to be very efficient for CG methods in [11].

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq C_k + \delta \alpha_k g_k^T d_k, \\ g_{k+1}^T d_k &\geq \sigma g_k^T d_k, \end{aligned} \quad (4)$$

where the definition of  $C_k$  is as follows:

$$C_0 = f(x_0), Q_0 = 1, Q_{k+1} = Q_k + 1, C_{k+1} = \frac{Q_k C_k + f_{k+1}}{Q_{k+1}}. \quad (5)$$

From Equation (5), we can find that  $C_k$  is a convex combination of the function values  $f(x_0)$  to  $f(x_k)$ . The generalized line search is non-monotonic, which facilitates the establishment of the global convergence of the algorithm under milder conditions.

Subspace technology plays an extraordinary role in solving large-scale unconstrained optimization problems. As the scale of optimization problems to be dealt with continues to expand, subspace technology has attracted increased attention from researchers. Using subspace minimization technology with CG method, Yuan and Stoer [12] creatively proposed the SMCG method, in which the approximate function of  $f(x)$  is minimized on the subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k\}$ , and the expression of the search direction is derived:

$$d_{k+1} = \mu_k g_{k+1} + \nu_k s_k, \quad (6)$$

where  $\mu_k$  and  $\nu_k$  are parameters, and  $s_k = x_{k+1} - x_k$ . Obviously, the SMCG method is a further promotion based on the CG method, and at the same time, it has a profound influence on the subsequent vigorous development of subspace technology. Based on Yuan's ideas above, Andrei [13] developed a new SMCG method, in which it further expands the search direction, develops into three subspaces, and used the acceleration strategy. Inspired by Andrei, Yang et al. [14] applied the technique of subspace minimization to another special three-term subspace and came up with a new SMCG method. On the same subspace, Li et al. [10] conducted a more in-depth study of Yang's results, analyzed more complex three-parameter situations, and set different conditions to dynamically select the search direction under different dimensions of subspace. Subspace technology has more extensive applications. Dai [15] proposed a new method called BBCG by fusing it with the Barzilai–Borwein [16] method and compared the performance of several BBCG methods proposed in the article through numerical experiments. It was found that the BBCG3 method has better performance. Many scholars have also tried to integrate the idea of minimizing subspace into the trust region method. For related research, readers can refer to [17]. More research on the use of subspace technology to construct different methods is still in progress [18–22].

The outline of this article is as follows: in Section 2, we give preliminary information. In Section 3, the search direction is discussed first, and then the obtained algorithm is proposed. Based on the above-mentioned work, under some mild assumptions, the global convergence of the algorithm for general functions is proved; more importantly, the result of R-linear convergence for uniformly convex functions is also established. Some numerical results for solving unconstrained optimization problems and image restoration problems are shown in Section 4. The conclusion and discussion are presented in Section 5.

## 2. Preliminary

The main work of this section is: in the subspace  $\Omega_{k+1} = \text{Span}\{-g_{k+1}, s_k, g_k\}$ , according to the different dimensions of  $\Omega_{k+1}$ , the discussion is divided into three cases; then, combined with the technique of subspace minimization, four forms of  $d_k$  are determined, and the conditions for dynamic selection of each direction are given.

In this paper, the direction at  $x_{k+1}$  is expected to minimize the quadratic approximation of the objective function,

$$\min_{d \in \Omega_{k+1}} \phi_{k+1}(d) = g_{k+1}^T d + \frac{1}{2} d^T B_{k+1} d, \tag{7}$$

on the subspace  $\Omega_{k+1}$ , where  $B_{k+1}$  is regarded as an approximation of the Hessian matrix and is positive definite. Assuming  $B_{k+1}$  satisfies the modified secant equation [23]

$$B_{k+1} s_k = y_k^* = y_k + \frac{\max\{z_k, 0\}}{s_k^T s_k} s_k, \tag{8}$$

where  $z_k = 2(f_k - f_{k+1}) + (g_{k+1} + g_k)^T s_k$ . Combined with (4), obviously,  $s_k^T y_k^* \geq s_k^T y_k > 0$ .

### 3. Proposed Method

#### 3.1. Direction Selection

According to the above discussion, as is known, the subspace may have three different dimensions; based on that, we analyze the selection of the search direction in the next section.

**Case I:**  $\dim(\Omega_{k+1}) = 3$ .

In this case, the direction can be expressed as

$$d_{k+1} = a_k g_{k+1} + b_k s_k + c_k g_k, \tag{9}$$

where  $a_k, b_k, c_k$  are parameters to be determined. Substituting (9) into (7), we get

$$\{a_k, b_k, c_k\} = \arg \min_{(a,b,c)} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T g_k \end{pmatrix}^T \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \frac{1}{2} \begin{pmatrix} a \\ b \\ c \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k^* & w_k \\ g_{k+1}^T y_k^* & s_k^T y_k^* & g_k^T y_k^* \\ w_k & g_k^T y_k^* & \rho_k \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \tag{10}$$

where  $\rho_k = g_k^T B_{k+1} g_k, \rho_{k+1} = g_{k+1}^T B_{k+1} g_{k+1}, w_k = g_{k+1}^T B_{k+1} g_k$ . Inspired by the BCG method [11], we set

$$w_k = \zeta_k \frac{g_{k+1}^T g_k \|y_k\|^2}{s_k^T y_k}, \zeta_k = \begin{cases} \max\{0.9\zeta_{k-1}, 1.2\}, & \text{if } \alpha_k > 1, \\ \min\{1.1\zeta_{k-1}, 1.75\}, & \text{otherwise,} \end{cases} \tag{11}$$

where  $\zeta_k$  is an adaptive parameter, and its value remains the same throughout the whole paper. Setting  $\zeta_0 = 1.5$ , we not only find that  $1.2 \leq \zeta_k \leq 1.75$ , but we also show that its numerical performance is better than a constant. The matrix in (10) is represented by  $D_k$ .

The positive definiteness of  $D_k$  is presented in Lemma 1. Now that we assume that  $D_k$  is positive definite, the unique solution of (10) can be calculated as follows:

$$\begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix} = -\frac{1}{\Delta_k} \begin{pmatrix} \chi & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & \nu \end{pmatrix} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T g_k \end{pmatrix}, \tag{12}$$

where

$$\begin{aligned} \Delta_k &= |D_k| = \rho_{k+1} \chi + w_k \theta_2 + g_{k+1}^T y_k^* \theta_1, \\ \chi &= \rho_k (s_k^T y_k^*) - (g_k^T y_k^*)^2, \\ \nu &= \rho_{k+1} (s_k^T y_k^*) - (g_{k+1}^T y_k^*)^2, \\ \theta_3 &= w_k (g_{k+1}^T y_k^*) - \rho_{k+1} (g_k^T y_k^*), \\ \theta_1 &= w_k (g_k^T y_k^*) - \rho_k (g_{k+1}^T y_k^*), \\ \theta &= \rho_k \rho_{k+1} - (w_k)^2, \\ \theta_2 &= (g_k^T y_k^*) (g_{k+1}^T y_k^*) - w_k (s_k^T y_k^*). \end{aligned}$$

If  $D_k$  is positive definite, then  $\Delta_k > 0$ , so

$$\rho_{k+1} \geq \frac{-w_k\theta_2 - g_{k+1}^T y_k^* \theta_1}{\chi} = h_k. \tag{13}$$

Setting  $n_k = 1 - \frac{(g_k^T y_k^*)^2}{\rho_k (s_k^T y_k^*)}$  and substituting the variable value in Equation (12), we get

$$\begin{aligned} \chi &= n_k \rho_k (s_k^T y_k^*), \\ h_k &= \frac{\left( \frac{w_k^2}{\rho_k} + \frac{(g_{k+1}^T y_k^*)^2}{s_k^T y_k^*} - \frac{2w_k (g_{k+1}^T y_k^*) (g_k^T y_k^*)}{\rho_k (s_k^T y_k^*)} \right)}{n_k}. \end{aligned} \tag{14}$$

Considering the formula in Equation (13), we compute  $\rho_{k+1}$  as

$$\rho_{k+1} = \zeta_k \max\{h_k, N\}, \tag{15}$$

where  $N = N_1 \|g_{k+1}\|^2$ ,  $N_1 = \max\left\{\frac{\|y_k^*\|^2}{s_k^T y_k^*}, \frac{4\|y_k^*\|^4 \|g_k^*\|^2}{\rho_k (s_k^T y_k^*)^2}\right\}$ .

In order to make the algorithm perform better, in a manner similar to [7,24], we set the following conditions:

$$\rho_0 \leq n_k, \tag{16}$$

$$\zeta_1 \leq \frac{s_k^T y_k}{\|s_k\|^2} \leq \frac{\|y_k^*\|^2}{s_k^T y_k^*} \leq \zeta_2, \tag{17}$$

$$\zeta_1 \leq \frac{\rho_k}{\|g_k\|^2}, \frac{4\|y_k^*\|^4 \|g_k^*\|^2}{\rho_k (s_k^T y_k^*)^2} \leq \zeta_2, \tag{18}$$

where  $\zeta_1, \zeta_2$  are positive constants,  $\rho_0 \in (0, 1)$ .

Now, we prove that  $D_k$  is positive definite.

**Lemma 1.** *If  $\rho_{k+1}$  is calculated by Equation (15), then the matrix  $D_k$  is positive definite.*

**Proof.** Using mathematical induction and  $\rho_0 \in (0, 1)$ , we can get  $\rho_{k+1} \geq \zeta_k N > 0$ ; because  $\rho_{k+1} > \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*}$ , so  $v = \rho_{k+1} (s_k^T y_k^*) - (g_{k+1}^T y_k^*)^2 > 0$ ; since  $\rho_{k+1} > h_k = \frac{-w_k\theta_2 - g_{k+1}^T y_k^* \theta_1}{\chi}$ , therefore  $\Delta_k = |D_k| = \rho_{k+1}\chi + w_k\theta_2 + g_{k+1}^T y_k^* \theta_1 > 0$ .  $\square$

**Case II:**  $\dim(\Omega_{k+1}) = 2$ .

We define  $d_k$  as

$$d_{k+1} = a_k g_{k+1} + b_k s_k, \tag{19}$$

where  $a_k, b_k$  are parameters. Similarly, substituting Equation (19) into the approximation function (7), we find

$$\{a_k, b_k\} = \arg \min_{(a,b)} \left( \begin{array}{c} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \end{array} \right)^T \begin{pmatrix} a \\ b \end{pmatrix} + \frac{1}{2} \begin{pmatrix} a \\ b \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k^* \\ g_{k+1}^T y_k^* & s_k^T y_k^* \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}. \tag{20}$$

If  $\Delta_k = \rho_{k+1} (s_k^T y_k^*) - (g_{k+1}^T y_k^*)^2 > 0$ , then Equation (20) has a unique solution:

$$\begin{pmatrix} a_k \\ b_k \end{pmatrix} = \frac{1}{\Delta_k} \begin{pmatrix} (g_{k+1}^T y_k^*) (g_{k+1}^T s_k) - (s_k^T y_k^*) \|g_{k+1}\|^2 \\ (g_{k+1}^T y_k^*) \|g_{k+1}\|^2 - \rho_{k+1} (g_{k+1}^T s_k) \end{pmatrix}. \tag{21}$$

Similar to the way that Equation  $w_k$  in (11) is evaluated, we set  $\rho_{k+1} = \zeta_k \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*}$ ; apparently,  $\Delta_k > 0$ . Furthermore, for the better performance of the algorithm, we require relevant variables to satisfy the condition in Equation (17).

As we all know, DY and HS methods have some good properties. For example, the finite termination of HS is helpful to improve the convergence rate. In view of the above considerations, we put forward an idea; when the conditions

$$\zeta_1 \leq \frac{s_k^T y_k}{\|s_k\|^2}, \frac{\|g_{k+1}\| \|d_k\|}{d_k^T y_k} \leq \zeta_3, \quad (22)$$

$$\frac{|g_{k+1}^T y_k g_{k+1}^T d_k|}{d_k^T y_k \|g_{k+1}\|^2} \leq \zeta_3, \quad (23)$$

are met, we take

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \beta_k = \max\{\beta_k^{HS}, \beta_k^{DY}\}, \quad (24)$$

where  $\zeta_3 \in [0, 1)$ .

Above all, in the case of two-dimensional subspace, when the condition (17) is established,  $d_k$  takes Equations (19) and (21); when the inequalities in Equations (22) and (23) are true,  $d_k$  is calculated by Equation (24).

**Case III:**  $\dim(\Omega_{k+1}) = 1$ .

When  $\dim(\Omega_{k+1}) = 1$ , we adopt the method of the steepest descent, namely  $d_{k+1} = -g_{k+1}$ .

### 3.2. Description of DSCG Algorithm

In this section, we first introduce an acceleration strategy (Algorithm 1) [25] which has been shown to be quite efficient for the CG method. Then, we present our dynamically adjusted subspace conjugate gradient algorithm (DSCG, Algorithm 2) and prove that the direction satisfies sufficient descent.

---

#### Algorithm 1: Acceleration Strategy.

---

Step 1: Compute:  $z = x_k + \alpha_k d_k$ ,  $g_z = \nabla f(z)$  and  $y_z = g_k - g_z$ ;

Step 2: Compute:  $\bar{a}_k = \alpha_k g_k^T d_k$  and  $\bar{b}_k = -\alpha_k y_k^T d_k$ ;

Step 3: If  $\bar{b}_k > 0$ , then, compute  $\eta_k = -\frac{\bar{a}_k}{\bar{b}_k}$  and update the variables as

$$x_{k+1} = x_k + \eta_k \alpha_k d_k, \text{ otherwise update the variables as } x_{k+1} = x_k + \alpha_k d_k.$$


---

---

#### Algorithm 2: DSCG.

---

Step 1: Given  $x_0 \in \mathbb{R}^n$ ,  $\alpha_0^{(0)}$ ,  $\epsilon > 0$ ,  $\zeta_1, \zeta_2, \zeta_3 \in [0, 1)$ . Let  $d_0 := -g_0$  and  $k := 0$ .

Step 2: When  $\|g_k\| \leq \epsilon$ , stop, otherwise go to step 3.

Step 3: Compute a stepsize  $\alpha_k$  that satisfies conditions (4) and (5), then adopt Algorithm 1 (acceleration strategy).

Step 4: Compute the direction  $d_k$ .

- Step 4.1: If the conditions (17) holds, compute  $d_{k+1}$  by Equations (19) and (21), and go to Step 5;
- Step 4.2: If the conditions (22) and (23) holds, compute  $d_{k+1}$  by Equation (24), and go to Step 5;
- Step 4.3: If the conditions (16)–(18) hold, compute  $d_{k+1}$  by Equations (9) and (12), and go to Step 5;
- Step 4.4: Otherwise, set  $d_{k+1} = -g_{k+1}$ .

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

---

### 3.3. Convergence Analysis

In this subsection, we focus on the convergence properties of the proposed algorithm (DSCG). The sufficient descent condition is crucial for a gradient descent algorithm. In order to establish the sufficient descent condition for the DSCG method, we firstly introduce the following lemma.

**Lemma 2.** If  $d_{k+1}$  is generated by Equations (9) and (12) or by Equations (19) and (20), then

$$g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}, \tag{25}$$

holds.

**Proof.** If  $d_{k+1}$  is generated by Equations (19) and (20), we get

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\frac{\|g_{k+1}\|^4}{\Delta_k} [s_k^T y_k^* - 2 \frac{(g_{k+1}^T y_k^*)(g_{k+1}^T s_k)}{\|g_{k+1}\|^2} + \rho_{k+1} (\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2})^2] \\ &\leq -\frac{\|g_{k+1}\|^4}{\Delta_k} H_{\min}(\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}) \\ &= -\frac{\|g_{k+1}\|^4}{\Delta_k} H(\frac{g_{k+1}^T y_k^*}{\rho_{k+1}}) \\ &= -\frac{\|g_{k+1}\|^4}{\Delta_k} \frac{\Delta_k}{\rho_{k+1}} \\ &= -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}, \end{aligned} \tag{26}$$

where  $H_{\min}(\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2})$  represents the minimum value of function  $s_k^T y_k^* - 2 \frac{(g_{k+1}^T y_k^*)(g_{k+1}^T s_k)}{\|g_{k+1}\|^2} + \rho_{k+1} (\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2})^2$  with  $\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}$  as the variable.

When  $d_{k+1}$  is given by Equations (9) and (12),

$$\begin{aligned} g_{k+1}^T d_{k+1} &= \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T g_k \end{pmatrix}^T \begin{pmatrix} a \\ b \\ c \end{pmatrix} \\ &= -\frac{1}{\Delta_k} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T g_k \end{pmatrix} \begin{pmatrix} \chi & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & \nu \end{pmatrix} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T g_k \end{pmatrix} \\ &= -\frac{\|g_{k+1}\|^4}{\Delta_k} \varphi(x, y), \end{aligned} \tag{27}$$

taking  $x = \frac{g_{k+1}^T g_k}{\|g_{k+1}\|^2}, y = \frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}$  as independent variable, where  $\varphi(x, y) = \nu x^2 + 2\theta_3 xy + \theta y^2 + 2\theta_2 x + 2\theta_1 y + \chi$ .

From Lemma 1, we find that  $\nu > 0, \nu\theta - \theta_3^2 = \Delta_k \rho_{k+1} > 0$ ; that is, there is a minimum value for  $\varphi(x, y)$ , which is calculated to be  $\varphi(x, y)_{\min} = \frac{\Delta_k}{\rho_{k+1}}$ .

Therefore, we can also get  $g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}$ .  $\square$

**Lemma 3.** Suppose  $d_{k+1}$  is generated by the DSCG algorithm. Then, there is a constant  $c_3 > 0$ , such that

$$g_{k+1}^T d_{k+1} \leq -c_3 \|g_{k+1}\|^2. \tag{28}$$

**Proof.** Based on the form of direction, we analyze this in three cases:

Case I: if  $d_{k+1} = -g_{k+1}$ , let  $c_3 = \frac{1}{2}$ , and thus it is proved.

Case II: When the direction is binomial, the following information must be considered.

We first discuss the case where  $d_{k+1}$  is given by Equations (19) and (21); here, for  $\rho_{k+1} = \tilde{\zeta}_k \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*}$ , combined with the conditions (11), (17) and Lemma 2, the following results can be obtained:

$$\rho_{k+1} \leq 2\zeta_2 \|g_{k+1}\|^2, g_{k+1}^T d_{k+1} \leq -\frac{1}{2\zeta_2} \|g_{k+1}\|^2. \quad (29)$$

When  $d_k$  is determined by Equation (24), for  $\beta_k = \beta_k^{DY}$ , we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \\ &\leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 \|g_{k+1}\| \|d_k\|}{d_k^T y_k} \\ &\leq -(1 - \zeta_3) \|g_{k+1}\|^2, \end{aligned} \quad (30)$$

for  $\beta_k = \beta_k^{HS}$ ; similarly, using (23), the same result can be obtained.

Case III: When the direction is computed by Equations (9) and (12), considering Lemma 2, we first prove that  $\rho_{k+1}$  has an upper bound.

$$\begin{aligned} |h_k| &= \left| \frac{\left( \frac{w_k^2}{\rho_k} + \frac{(g_{k+1}^T y_k^*)^2}{s_k^T y_k^*} - \frac{2w_k(g_{k+1}^T y_k^*)(g_k^T y_k^*)}{\rho_k(s_k^T y_k^*)} \right)}{n_k} \right| \\ &\leq \left| \left( \frac{4\|y_k^*\|^4 \|g_k^*\|^2}{\rho_k(s_k^T y_k^*)^2} + \frac{\|y_k^*\|^2}{s_k^T y_k^*} \right) \|g_{k+1}\|^2 + 2 \frac{w_k}{\sqrt{\rho_k}} \frac{g_{k+1}^T y_k^*}{\sqrt{s_k^T y_k^*}} \frac{g_k^T y_k^*}{\sqrt{\rho_k s_k^T y_k^*}} \right| / n_k \\ &\leq (2N_1 \|g_{k+1}\|^2 + 2 \frac{|w_k| \|g_{k+1}^T y_k^*\|}{\sqrt{\rho_k} \sqrt{s_k^T y_k^*}}) / n_k \\ &\leq (2N + 2\sqrt{N}\sqrt{N}) / n_k \\ &\leq \frac{4N}{\rho_0} \end{aligned} \quad (31)$$

The above formula follows from Equations (11), (14), (15), and  $\chi > 0$ . By using the conditions in Equations (16) and (17), we have

$$\rho_{k+1} = \zeta_k \max\{h_k, N\} \leq 2 \frac{4N}{\rho_0} = \frac{8N_1 \|g_{k+1}\|^2}{\rho_0} \leq \frac{8\zeta_2 \|g_{k+1}\|^2}{\rho_0}. \quad (32)$$

Finally, using the conclusion of Lemma 2, it is concluded that

$$g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}} \leq -\frac{\rho_0}{8\zeta_2} \|g_{k+1}\|^2. \quad (33)$$

Summarizing all the above cases, we take

$$c_3 = \max\left\{ \frac{1}{2}, \frac{1}{2\zeta_2}, (1 - \zeta_3), \frac{\rho_0}{8\zeta_2} \right\}, \quad (34)$$

and thus complete the proof.  $\square$

In the remainder of this subsection, the global convergence of the algorithm for general functions is proved; more importantly, the result of R-linear convergence for uniformly convex functions is also established in this section. We first introduce two necessary assumptions.

**Assumption 1.** Function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differentiable and has a lower bound on  $\mathbb{R}^n$ .

**Assumption 2.** The gradient function  $g(x)$  is Lipschitz continuous with a constant  $L > 0$ ; i.e.,

$$\|g(x) - g(y)\| \leq L \|x - y\|, x, y \in \mathbb{R}^n, \quad (35)$$

which means that  $\|y_k\| \leq L\|s_k\|$ . Remarkably, Assumption 1 is milder than the usual assumption: the level set  $D = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$  is bounded.

**Lemma 4.** Supposing  $\alpha_k$  is generated by a generalized line search (4) and satisfies Assumption 2, then

$$\alpha_k \geq \frac{(1 - \sigma)|g_k^T d_k|}{L\|d_k\|^2}. \tag{36}$$

**Proof.** By the line search condition (4), we get

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

Since  $\sigma < 1$  and  $g_k^T d_k < 0$ , then (36) holds immediately.  $\square$

**Lemma 5.** If  $\alpha_k$  fulfills the generalized line search conditions (4), (5) and Assumption 1 holds, it follows that

$$f_k \leq C_k, \forall k. \tag{37}$$

**Proof.** From (4) and  $g_k^T d_k < 0$ , there is  $f_{k+1} \leq C_k$ . Condition (5) shows that

$$Q_{k+1}C_{k+1} = Q_k C_k + f_{k+1} \leq Q_k C_k + C_k = (Q_k + 1)C_k, \tag{38}$$

namely,  $C_{k+1} \leq C_k$ . Then,

$$Q_k C_k + f_{k+1} = Q_k C_{k+1} + C_{k+1} \leq Q_k C_k + C_{k+1}, \tag{39}$$

which means  $f_{k+1} \leq C_{k+1}$ .  $\square$

**Lemma 6.** Let  $d_{k+1}$  be generated by the DSCG algorithm. Then, there is a constant  $c_4 > 0$  such that

$$\|d_{k+1}\| \leq c_4 \|g_{k+1}\|. \tag{40}$$

**Proof.** Similarly, we analyze it in three cases.

Case I: If  $d_{k+1} = -g_{k+1}$ , let  $c_4 = 1$ , and thus it is proved.

Case II: If  $d_{k+1}$  is given by Equation (24), from Assumption 2 and condition (22),

$$\begin{aligned} \|d_{k+1}\| &= \| -g_{k+1} + \beta_k d_k \| \\ &\leq \|g_{k+1}\| + \frac{\|g_{k+1}\| \|y_k\| \|d_k\|}{d_k^T y_k} \\ &\leq (1 + \frac{L}{\zeta_1}) \|g_{k+1}\|. (\beta_k = \beta_k^{HS}) \end{aligned} \tag{41}$$

When  $\beta_k = \beta_k^{DY}$ , using the same method, we can get (41).

Now, if  $d_k$  is calculated by Equations (19) and (21), where  $\rho_{k+1} = \zeta_k \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*}$ , conditions (11) and (17) hold, then

$$\begin{aligned} \Delta_k &= \rho_{k+1}(s_k^T y_k^*) - (g_{k+1}^T y_k^*)^2 \\ &= s_k^T y_k^* (\rho_{k+1} - \frac{(g_{k+1}^T y_k^*)^2}{s_k^T y_k^*}) \\ &\geq s_k^T y_k^* (\frac{1}{5} \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*}) \\ &\geq \frac{1}{5} \zeta_1 \|s_k\|^2 \frac{\|g_{k+1}\|^2 \|y_k^*\|^2}{s_k^T y_k^*} \end{aligned} \tag{42}$$



Applying the above results, combined with Cauchy inequality and triangle inequality, we have

$$\begin{aligned}
 \|d_{k+1}\| &= \|a_k g_{k+1} + b_k s_k\| \\
 &\leq \frac{1}{\Delta_k} (|g_{k+1}^T y_k^*| |g_{k+1}^T s_k| \|g_{k+1}\| + |(g_{k+1}^T y_k^*)| \|g_{k+1}\|^2 + \rho_{k+1} (g_{k+1}^T s_k) \|s_k\|) \\
 &\leq \frac{1}{\Delta_k} (2\|s_k\| \|y_k^*\| + \frac{\rho_{k+1} \|s_k\|^2}{\|g_{k+1}\|^2}) \|g_{k+1}\|^3 \\
 &\leq \frac{5s_k^T y_k^*}{\zeta_1 \|s_k\|^2 \|y_k^*\|^2} (2\|s_k\| \|y_k^*\| + \frac{\rho_{k+1} \|s_k\|^2}{\|g_{k+1}\|^2}) \|g_{k+1}\| \\
 &= (\frac{10s_k^T y_k^*}{\zeta_1 \|s_k\| \|y_k^*\|} + \frac{5\zeta_k}{\zeta_1}) \|g_{k+1}\| \\
 &\leq (\frac{10s_k^T y_k^*}{\zeta_1 \|s_k\| \|y_k^*\|} + \frac{10}{\zeta_1}) \|g_{k+1}\| \\
 &\leq \frac{20}{\zeta_1} \|g_{k+1}\|.
 \end{aligned} \tag{43}$$

Case III: When  $d_k$  is computed by Equations (9) and (12), Similarly, let us first discuss a lower bound of  $\Delta_k$ . Based on Equations (11), (13), and (15), we have

$$\begin{aligned}
 \Delta_k &= \rho_{k+1}\chi + w_k\theta_2 + g_{k+1}^T y_k^* \theta_1 \\
 &= \chi(\rho_{k+1} - \frac{-w_k\theta_2 - g_{k+1}^T y_k^* \theta_1}{\chi}) \\
 &= \chi(\rho_{k+1} - h_k) \\
 &\geq \chi(1.2\max\{h_k, N\} - h_k) \\
 &\geq \frac{1}{5}\chi N.
 \end{aligned} \tag{44}$$

Defining  $\chi_1 = \rho_k (s_k^T y_k^*)$ , from Equations (14) and (16), we have  $\chi = \chi_1 n_k \geq \chi_1 \rho_0 > 0$ ; therefore,

$$\Delta_k \geq \frac{1}{5}\chi_1 \rho_0 N. \tag{45}$$

Then,

$$\begin{aligned}
 \|d_{k+1}\| &= \|a_k g_{k+1} + b_k s_k + c_k g_k\| \\
 &= \frac{1}{\Delta_k} \begin{pmatrix} \|g_{k+1}\|^2 \\ |g_{k+1}^T s_k| \\ |g_{k+1}^T g_k| \end{pmatrix} \begin{pmatrix} |\chi| & |\theta_1| & |\theta_2| \\ |\theta_1| & |\theta| & |\theta_3| \\ |\theta_2| & |\theta_3| & |\nu| \end{pmatrix} \begin{pmatrix} \|g_{k+1}\| \\ \|s_k\| \\ \|g_k\| \end{pmatrix} \\
 &\leq \frac{\|g_{k+1}\|}{\Delta_k} (\chi_1 \|g_{k+1}\|^2 + 4e_k \sqrt{\chi_1 N_1} \|g_{k+1}\|^2 + 2i_k \sqrt{\chi_1} (N + \rho_{k+1}) + j_k \rho_{k+1}) \\
 &\leq \frac{5\|g_{k+1}\|}{\chi_1 \rho_0 N} (\chi_1 \|g_{k+1}\|^2 + 4e_k \sqrt{\chi_1 N_1} \|g_{k+1}\|^2 + 2i_k \sqrt{\chi_1} (N + \rho_{k+1}) + j_k \rho_{k+1}) \\
 &\leq \|g_{k+1}\| (\frac{5}{\rho_0 N_1} + \frac{20}{\rho_0 \sqrt{N_1}} \frac{e_k}{\sqrt{\chi_1}} + \frac{10(\rho_0 + 8)}{(\rho_0)^2} \frac{i_k}{\sqrt{\chi_1}} + \frac{40}{(\rho_0)^2} \frac{j_k}{\chi_1}),
 \end{aligned} \tag{46}$$

where  $e_k = \rho_k \|s_k\| + \sqrt{s_k^T y_k^*} \|g_k\|$ , and  $i_k = \|s_k\| \|g_k\|$ ,  $j_k = \rho_k \|s_k\|^2 + s_k^T y_k^* \|g_k\|^2$ . From Equations (15), (17), and (18), we obtain

$$\zeta_1 \leq \frac{\|y_k^*\|^2}{s_k^T y_k^*} \leq N_1,$$

$$\begin{aligned}\frac{e_k}{\sqrt{\chi_1}} &= \frac{\|s_k\|}{\sqrt{s_k^T y_k^*}} + \frac{\|g_k\|}{\sqrt{\rho_k}} \leq \frac{2}{\sqrt{\zeta_1}}, \\ \frac{i_k}{\sqrt{\chi_1}} &= \frac{\|s_k\| \|g_k\|}{\sqrt{\rho_k (s_k^T y_k^*)}} \leq \frac{1}{\zeta_1}, \\ \frac{j_k}{\chi_1} &= \frac{\|s_k\|^2}{s_k^T y_k^*} + \frac{\|g_k\|^2}{\rho_k} \leq \frac{1}{\zeta_1} + \frac{1}{\zeta_1} = \frac{2}{\zeta_1}.\end{aligned}$$

Based on the above results, it can be deduced further that

$$\begin{aligned}\|d_{k+1}\| &\leq \left( \frac{5}{\rho_0 \zeta_1} + \frac{40}{\rho_0 \zeta_1} + \frac{10(\rho_0 + 8)}{(\rho_0)^2} \frac{1}{\zeta_1} + \frac{40}{(\rho_0)^2} \frac{2}{\zeta_1} \right) \|g_{k+1}\| \\ &\leq \left( \frac{55\rho_0 + 160}{\rho_0^2 \zeta_1} \right) \|g_{k+1}\|.\end{aligned}\quad (47)$$

In conclusion, let

$$c_4 = \min\left\{1 + \frac{L}{\zeta_1}, \frac{20}{\zeta_1}, \frac{55\rho_0 + 160}{\rho_0^2 \zeta_1}\right\}, \quad (48)$$

and thus (40) holds.  $\square$

**Theorem 1.** Assuming that Assumptions 1 and 2 hold, the sequence  $\{x_k\}$  is generated by the DSCG algorithm, and we have

$$\lim_{k \rightarrow \infty} \inf \|g_k\| = 0. \quad (49)$$

**Proof.** According to the generalized line search conditions (4) and (5), we know that

$$f_{k+1} \leq C_k + \delta \alpha_k g_k^T d_k, \quad Q_{k+1} = Q_k + 1, \quad C_{k+1} = \frac{Q_k C_k + f_{k+1}}{Q_{k+1}}, \quad (50)$$

Combined with Lemmas 3, 4, and 6, it follows that

$$\begin{aligned}f_{k+1} &\leq C_k - \frac{(1-\sigma)\delta}{L} \left( \frac{g_k^T d_k}{\|d_k\|} \right)^2 \\ &\leq C_k - \frac{(1-\sigma)\delta c_3^2}{L c_4^2} \|g_k\|^2 \\ &= C_k - \beta \|g_k\|^2 \quad (\beta = \frac{(1-\sigma)\delta c_3^2}{L c_4^2}).\end{aligned}\quad (51)$$

Since

$$Q_{k+1} = Q_k + 1 = k + 2,$$

therefore,

$$C_{k+1} \leq \frac{Q_k C_k + C_k - \beta \|g_k\|^2}{Q_{k+1}} = C_k - \frac{\beta \|g_k\|^2}{Q_{k+1}}. \quad (52)$$

According to Assumption 1 and Lemma 5, it can be seen that  $C_{k+1}$  has a lower bound, and so

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{Q_{k+1}} < \infty. \quad (53)$$

Thus, we proved that Equation (49) holds.  $\square$

**Theorem 2.** *Supposing that Assumptions 1 and 2 hold,  $f$  is a uniformly convex function, and the unique minimizer is  $x^*$ , the sequence  $\{x_k\}$  is generated by the DSCG algorithm. For all  $k$ , there exists  $\hat{\alpha} > 0$  such that  $\alpha_k \leq \hat{\alpha}$ ,  $\tau_{max} < 1$ . Then, there is a constant  $\theta \in (0, 1)$ , which makes*

$$f_k - f(x^*) \leq \theta^k(f_0 - f(x^*)). \tag{54}$$

**Proof.** In the proof of Lemma 5, we know that  $C_k > C_{k+1} > f(x^*)$ , which implies that

$$0 < \frac{C_{k+1} - f(x^*)}{C_k - f(x^*)} < 1, \forall k \geq 0. \tag{55}$$

Denote  $r = \limsup_{k \rightarrow \infty} \frac{C_{k+1} - f(x^*)}{C_k - f(x^*)}$ ; obviously,  $r \in [0, 1]$ .

Let us first analyze the case when  $r = 1$ . Now, there is a subsequence  $\{x_{k_j}\}$  such that

$$\lim_{j \rightarrow \infty} \frac{C_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} = 1. \tag{56}$$

From Equation (2.15) of [26], we know that  $0 < 1 - \tau_{max} \leq \frac{1}{Q_k} \leq 1$ . Thus, there exists a convergent subsequence  $\{\frac{1}{Q_{k_j+1}}\}$ . We assume that

$$\lim_{j \rightarrow \infty} \frac{1}{Q_{k_j} + 1} = r_1, \tag{57}$$

and so  $0 < r_1 \leq 1$ .

Through the expression of  $C_{k+1}$  (1.6), we have

$$\frac{C_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} = \left(1 - \frac{1}{Q_{k_j} + 1}\right) + \frac{1}{Q_{k_j} + 1} \frac{f_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)}. \tag{58}$$

Combining the above three formulas, it is obvious that

$$\lim_{j \rightarrow \infty} \frac{f_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} = 1. \tag{59}$$

Based on Equation (3.4) of [26], we know that the uniformly convex function  $f$  has the following property:

$$f_{k_j+1} - f(x^*) \leq \gamma \|g_{k_j+1}\|^2 \tag{60}$$

It is known that  $\alpha_k \leq \hat{\alpha}$ ,  $g$  is Lipschitz continuous; thus, it follows that

$$\begin{aligned} \|g_{k_j+1}\| &\leq \|g_{k_j+1} - g_{k_j}\| + \|g_{k_j}\| \\ &\leq L \|x_{k_j+1} - x_{k_j}\| + \|g_{k_j}\| \\ &= L \alpha_{k_j} \|d_{k_j}\| + \|g_{k_j}\| \\ &\leq (1 + L \hat{\alpha} c_4) \|g_{k_j}\| \end{aligned} \tag{61}$$

Therefore,  $f_{k_j+1} - f(x^*) \leq \gamma(1 + L \hat{\alpha} c_4)^2 \|g_{k_j}\|^2$ , and

$$0 < \frac{f_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} \leq \frac{\gamma(1 + L \hat{\alpha} c_4)^2 \|g_{k_j}\|^2}{C_{k_j} - f(x^*)}. \tag{62}$$

Condition (51) means that

$$f_{k_j+1} - f(x^*) \leq C_{k_j} - f(x^*) - \beta \|g_{k_j}\|^2, \tag{63}$$

and

$$\frac{f_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} \leq 1 - \frac{\beta \|g_{k_j}\|^2}{C_{k_j} - f(x^*)}. \tag{64}$$

From Equation (59), we can see that

$$\lim_{j \rightarrow \infty} \frac{\|g_{k_j}\|^2}{C_{k_j} - f(x^*)} = 0. \tag{65}$$

Combined with condition (62),

$$\lim_{j \rightarrow \infty} \frac{f_{k_j+1} - f(x^*)}{C_{k_j} - f(x^*)} = 0 \tag{66}$$

Obviously, this conflicts with Equation (59), so  $r \neq 1$ ; i.e.,

$$\limsup_{k \rightarrow \infty} \frac{C_{k+1} - f(x^*)}{C_k - f(x^*)} = r < 1 \tag{67}$$

Thus, there is an integer  $k_0 > 0$  such that

$$\frac{C_{k+1} - f(x^*)}{C_k - f(x^*)} \leq r + \frac{1-r}{2} = \frac{1+r}{2} < 1, \forall k \geq k_0. \tag{68}$$

It is deduced that  $0 < \max_{0 \leq k \leq k_0} \left\{ \frac{C_{k+1} - f(x^*)}{C_k - f(x^*)} \right\} = \hat{r} < 1$ . Define  $\theta = \max\left\{ \frac{1+r}{2}, \hat{r} \right\}$ , from condition (55); obviously,

$$C_{k+1} - f(x^*) \leq \theta(C_k - f(x^*)), \quad 0 < \theta < 1. \tag{69}$$

It follows from condition (69) that

$$C_{k+1} - f(x^*) \leq \theta(C_k - f(x^*)) \leq \theta^{k+1}(C_0 - f(x^*)). \tag{70}$$

Lemma 5 ( $f_k \leq C_k$ ) and  $C_0 = f_0$  imply that (54) holds.  $\square$

#### 4. Numerical Results

In this section, we report the numerical performance of the DSCG algorithm from two aspects. Firstly, the algorithm is compared with TTS [13] and CG\_DESCENT [27] algorithms on the normal unconstrained problem; secondly, the algorithm is applied to the image restoration problem, and the numerical results are observed. The running environment of all codes is a PC with 2.20 GHz CPU, 4.00 GB RAM memory, and the Windows 10 operating system.

##### 4.1. Unconstrained Problem

The experiment selected 73 test functions, as shown in Table 1.

The dimensions of the function were set as  $E^n = 3000$ ,  $n = 6000$ , and  $n = 9000$ .

The iteration stop criterion is as follows: if  $\|g(x)\| < 10^{-6}$  or the number of iterations exceeds 1000 and  $stop1 < 10^{-5}$ , then the algorithm will terminate, where  $stop1 = \frac{|f_{k+1} - f_k|}{|f_k|}$ , when  $|f_k| > 10^{-5}$ ; otherwise,  $stop1 = |f_{k+1} - f_k|$ .

The parameters used by the algorithm are  $\delta = 0.1$ ,  $\sigma = 0.8$ ,  $\zeta_1 = 10^{-7}$ ,  $\zeta_2 = 10^5$ ,  $\zeta_3 = 10^{-5}$ ,  $\rho_0 = 0.8$ .

The initial stepsize selection strategy [27] is

$$\alpha_0^{(0)} = \begin{cases} 1.0, & \text{if } \|x_0\|_\infty < 10^{-30} \text{ and } \|f_0\|_\infty < 10^{-30}, \\ 2 \frac{\|f_0\|}{\|g_0\|}, & \text{if } \|x_0\|_\infty < 10^{-30} \text{ and } \|f_0\|_\infty \geq 10^{-30}, \\ \min\{1.0, \frac{\|x_0\|_\infty}{\|g_0\|_\infty}\}, & \text{if } \|x_0\|_\infty \geq 10^{-30} \text{ and } \|g_0\|_\infty < 10^7, \\ \min\{1.0, \max\{\frac{\|x_0\|_\infty}{\|g_0\|_\infty}, \frac{1}{\|g_0\|_\infty}\}\}, & \text{if } \|x_0\|_\infty \geq 10^{-30} \text{ and } \|g_0\|_\infty \geq 10^7, \end{cases}$$

where  $\|\cdot\|_\infty$  represents the infinite norm.

TTS and CG\_DESCENT use the parameters in their code. We apply the profiles of Dolan and Moré [28] to evaluate the effectiveness of the three algorithms and discuss the performance profiles of the algorithm in CPU, NFG, and NI in detail.

The meanings of some symbols in the text are as follows:

**N0:** The serial number of the test problem;

**CPU:** The running time of algorithm (seconds);

**NFG:** Total evaluation numbers of function and gradient;

**NI:** The number of iterations.

**Table 1.** The test problems.

Test Problems	No.	Test Problems	No.
Extended Freudenstein and Roth Function	1	ARWHEAD Function (CUTE)	38
Extended Trigonometric Function	2	ARWHEAD Function (CUTE)	39
Extended Rosenbrock Function	3	NONDQUAR Function (CUTE)	40
Extended White and Holst Function	4	DQDRTIC Function (CUTE)	41
Extended Beale Function	5	EG2 Function (CUTE)	42
Extended Penalty Function	6	DIXMAANA Function (CUTE)	43
Perturbed Quadratic Function	7	DIXMAANB Function (CUTE)	44
Raydan 1 Function	8	DIXMAANC Function (CUTE)	45
Raydan 2 Function	9	DIXMAANE Function (CUTE)	46
Diagonal 1 Function	10	Partial Perturbed Quadratic Function	47
Diagonal 2 Function	11	Broyden Tridiagonal Function	48
Diagonal 3 Function	12	Almost Perturbed Quadratic Function	49
Hager Function	13	Tridiagonal Perturbed Quadratic Function	50
Generalized Tridiagonal 1 Function	14	EDENSCH Function (CUTE)	51
Extended Tridiagonal 1 Function	15	VARDIM Function (CUTE)	52
Extended Three Exponential Terms Function	16	STAIRCASE S1 Function	53
Generalized Tridiagonal 2 Function	17	LIARWHD Function (CUTE)	54
Diagonal 4 Function	18	DIAGONAL 6 Function	55
Diagonal 5 Function	19	DIXON3DQ Function (CUTE)	56
Extended Himmelblau Function	20	DIXMAANF Function (CUTE)	57
Generalized PSC1 Function	21	DIXMAANG Function (CUTE)	58
Extended PSC1 Function	22	DIXMAANH Function (CUTE)	59
Extended Powell Function	23	DIXMAANI Function (CUTE)	60
Extended Block Diagonal BD1 Function	24	DIXMAANJ Function (CUTE)	61
Extended Maratos Function	25	DIXMAANK Function (CUTE)	62
Extended Cliff Function	26	IXMAANL Function (CUTE)	D63
Quadratic Diagonal Perturbed Function	27	DIXMAAND Function (CUTE)	64
Extended Wood Function	28	ENGVAL1 Function (CUTE)	65
Extended Hiebert Function	29	FLETCHCR Function (CUTE)	66
Quadratic Function QF1 Function	30	COSINE Function (CUTE)	67
Extended Quadratic Penalty QP1 Function	31	Extended DENSCHNB Function (CUTE)	68
Extended Quadratic Penalty QP2 Function	32	DENSCHNF Function (CUTE)	69
A Quadratic Function QF2 Function	33	SINQUAD Function (CUTE)	70
Extended EP1 Function	34	BIGGSB1 Function (CUTE)	71
Extended Tridiagonal-2 Function	35	Partial Perturbed Quadratic PPQ2 Function	72
BDQRTIC Function (CUTE)	36	Scaled Quadratic SQ1 Function	73
TRIDIA Function (CUTE)	37		

When the problem dimension is 9000, the CG\_DESCENT method only solves 64 problems, while the other two methods successfully complete all the problems. Compared with other methods, the method represented by the top curve in a performance profile drawing can solve the most problems in the best time range.

As shown in Figure 1, it is clear that the DSCG method is superior to other algorithms in terms of CPU time. This corresponds to the top curve and can solve 47.49% of the test problems in the shortest time. In contrast, TTS is the fastest at solving 40.64% of the test problems, and CG\_DESCENT is the fastest for only 9.59% of problems.

Now, let us focus on Figure 2. By comparison, it is found that DSCG requires fewer functions and gradient evaluations than other algorithms, which helps to simplify calculation and improve algorithm efficiency. It can solve 58.9% of the test problems with minimal function and gradient evaluations. TTS can solve 28.77% of the test problems with the least amount of function and gradient evaluations. The proportion of CG\_DESCENT corresponds to 13.24%.

In addition, Figure 3 shows the performance comparison results of each algorithm in terms of the number of iterations. It can be seen from the figure that the performance of the DSCG algorithm is outstanding, as it can solve 64.38% of the problems with the minimum number of iterations. At the same time, TTS and CG\_DESCENT have the least iterations in 52.05% and 7.76% of the problems, respectively.

The above three pictures of CPU, NFG, and NI contain some similar information. It is concluded that in the given test set, DSCG performs very well, with numerical results superior to those of TTS and CG\_DESCENT.

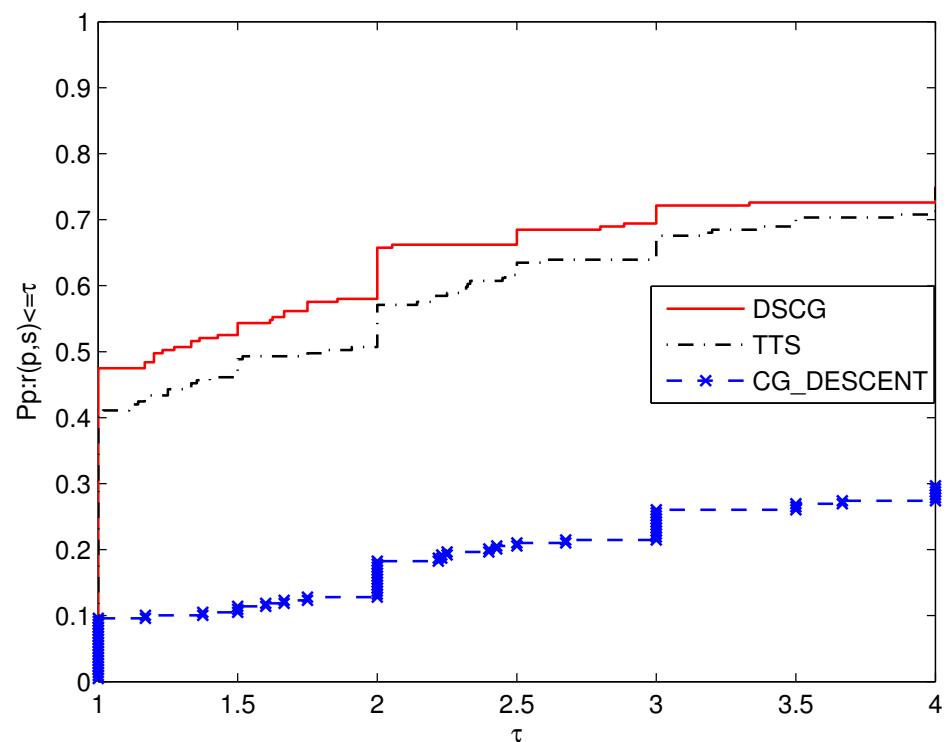


Figure 1. Performance profiles for the CPU.

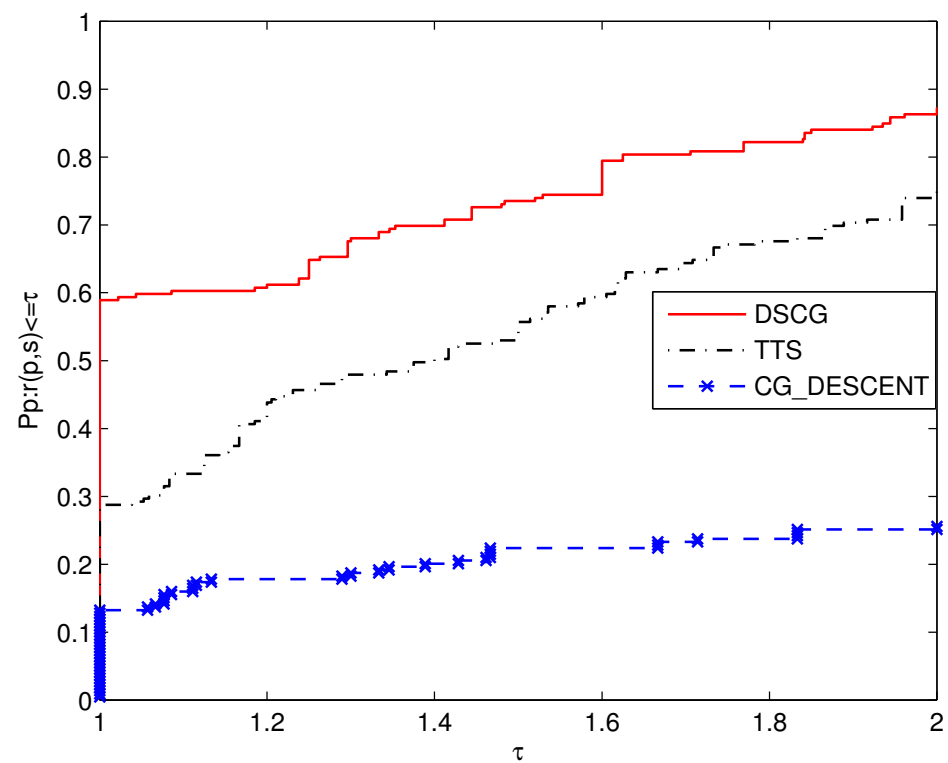


Figure 2. Performance profiles for the NFG.

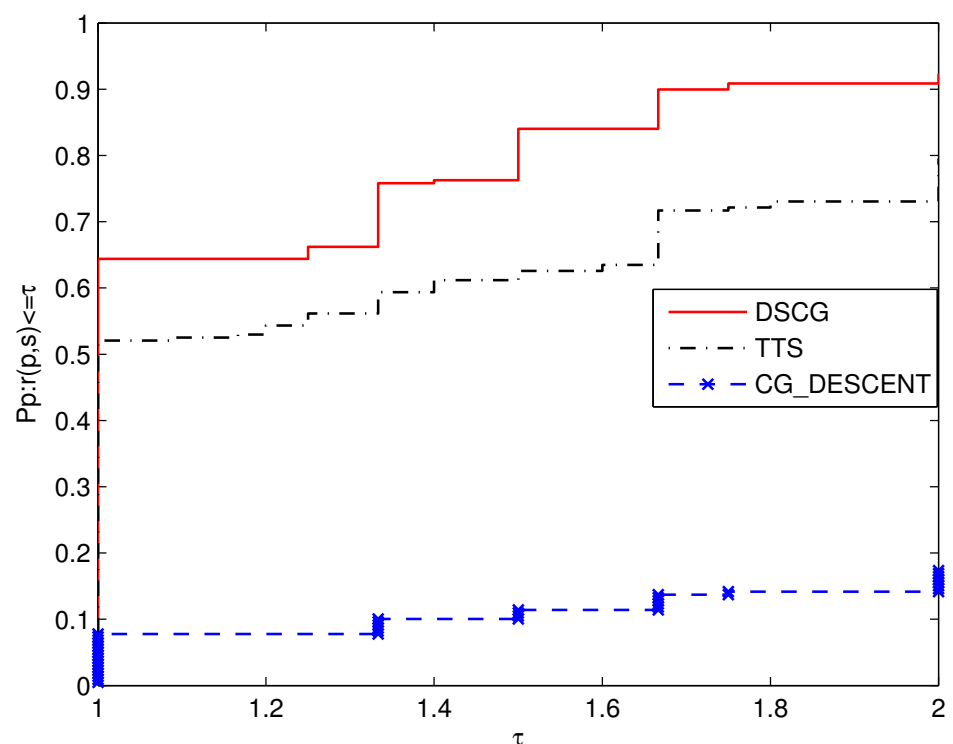


Figure 3. Performance profiles for the NI.

#### 4.2. Image Restoration Problem

The proposed DSCG is also applied to the problem of image restoration in this subsection. For more professional work in the field of image processing, please see [29,30]. In two scenes with different noise values, the blurred original image is repaired to make the picture clear and recognizable. This work has a wide range of applications in many fields

of production and life, with important practical significance, and is also a difficult subject in the field of optimization. Its basic model is  $b = Ax + \zeta$ , where  $x \in \mathbb{R}^n$  is the original image,  $A \in \mathbb{R}^{m \times n}$  is the blur matrix,  $\zeta \in \mathbb{R}^m$  represents noise, and  $b \in \mathbb{R}^m$  is the image observed after noise reduction. The unknown noise value  $\zeta$  is usually obtained through

$$\min_{x \in \mathbb{R}^n} \|Ax + b\|^2, \quad (71)$$

but because the image system is susceptible to noise and lack of information, it is difficult to obtain a satisfactory solution. In order to overcome the above shortcoming, the least square model is usually introduced,

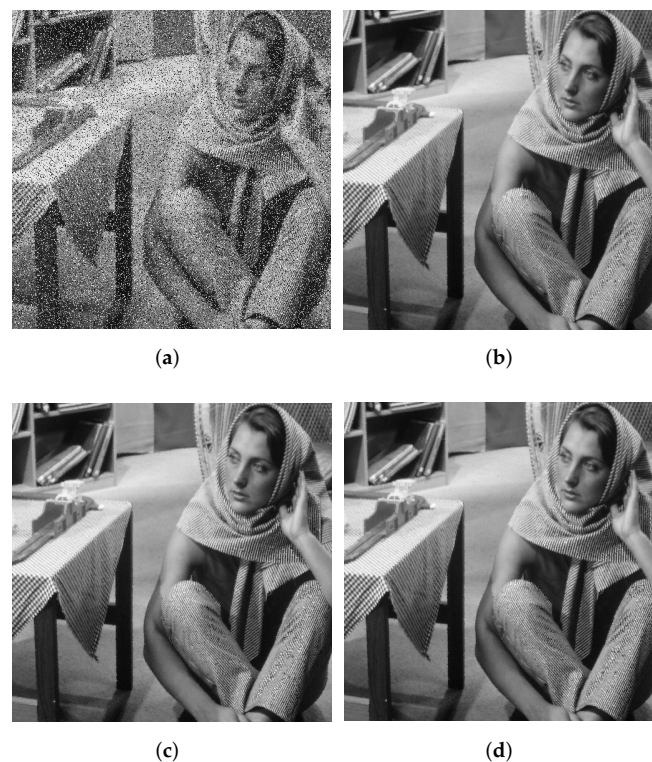
$$\min_{x \in \mathbb{R}^n} \|Ax + b\|^2 + \lambda \|Yx\|_1, \quad (72)$$

where  $Y$  is the linear operator,  $\|\cdot\|_1$  represents the  $l_1$  norm, and  $\lambda$  is the regularization parameter used to weigh the data item and the regularization term.

**Stop condition:**  $\frac{|f_{k+1} - f_k|}{|f_k|} < 10^{-3}$  or  $\frac{\|x_{k+1}\| - \|x_k\|}{\|x_k\|} < 10^{-3}$ ;

**Tested picture:** Barbara ( $512 \times 512$ ), Baboon ( $512 \times 512$ ).

The specific image repair results of the two algorithms under different noise values are shown in Figures 4 and 5. Obviously, for a given set of pictures, both algorithms successfully completed the repair. Let us focus on the comparison on Table 2, which presents the CPU time spent in the repair process of algorithms.

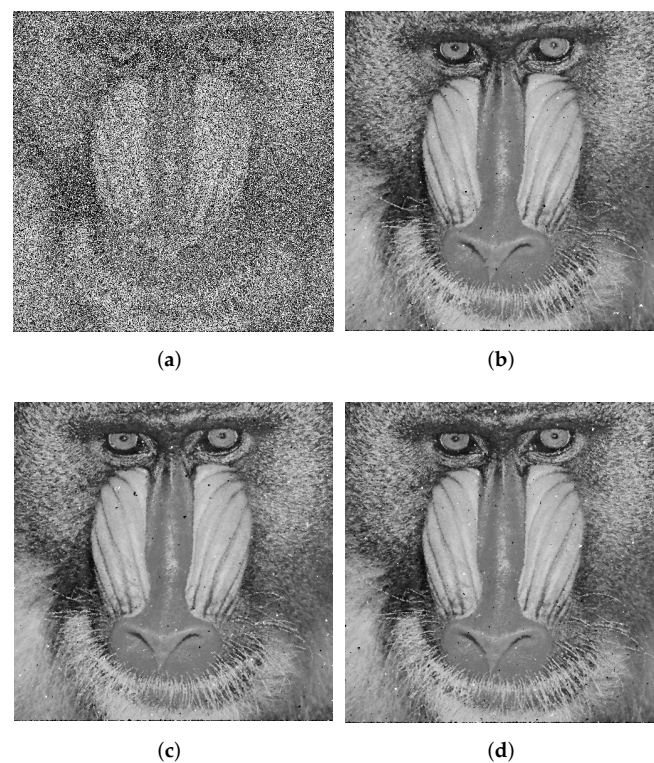


**Figure 4.** 20% noise. (a) original image; (b) DSCG; (c) TTS; (d) CG\_DESCENT.

Many works have focused on the image restoration problem, and more detailed references can be found in [31–34]. In this paper, we display the original image to be repaired, and the repaired results of DSCG, TTS, and CG\_DESCENT from left to right.

Summarizing the information contained in the pictures and tables in this section, we have obtained two conclusions: (i) both algorithms are capable of repairing pictures within a reasonable time frame; (ii) with noise interference of 20% and 50%, DSCG is shown to be a promising algorithm.





**Figure 5.** 50% noise. (a) original image; (b) DSCG; (c) TTS; (d) CG\_DESCENT.

**Table 2.** CPU time spent by algorithms (seconds).

20% Noise	Barbara	Baboon	Total
DSCG	1.2656	1.2813	2.5469
TTS	1.2813	1.2656	2.5469
CG_DESCENT	1.4219	1.2969	2.7188
50% Noise	Barbara	Baboon	Total
DSCG	2.0625	1.9688	4.0313
TTS	2.125	2.1719	4.2969
CG_DESCENT	2.0313	2.1875	4.2188

## 5. Conclusions and Discussion

In this paper, an algorithm for dynamically adjusting direction was proposed, which corresponds to the directions of four calculation forms by satisfying different conditions. We discuss the selection of directions in a special three-term subspace using modified secant equations, subspace minimization techniques, and acceleration strategies. The algorithm has a good property: each search direction satisfies the sufficient descent condition. We use the nonmonotonic generalized line search to obtain remarkable results: under some mild assumptions, we not only prove the global convergence of the general function algorithm but also further prove the R-linear convergence of the uniformly convex function. Interestingly, we apply this algorithm to image restoration, and the algorithm has good numerical performance in both the unconstrained and image restoration problems, which fully demonstrates the efficiency of this algorithm.

**Author Contributions:** Conceptualization, J.H. and S.Y.; methodology, S.Y.; software and validation, Y.W.; visualization and formal analysis, Y.W.; writing—original draft preparation, Y.W.; supervision, G.X. and S.Y.; funding acquisition, G.X. and S.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Natural Science Foundation of China No. 71862003, Natural Science Foundation of Guangxi Province (CN) No. 2020GXNSFAA159014, Program for Innovative Team of Guangxi University of Finance and Economics, and the Special Funds for Local Science and Technology Development Guided by the Central Government grant number ZY20198003.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Polyak, B.T. The conjugate gradient method in extreme problems. *USSR Comput. Math. Math. Phys.* **1969**, *9*, 94–112. [[CrossRef](#)]
2. Hestenes, M.R.; Stiefel, E.L. Methods of conjugate gradient for solving linear systems. *Res. Natl. Bur. Stand.* **1952**, *6*, 409–436. [[CrossRef](#)]
3. Liu, Y.L.; Storey, C.S. Efficient generalized conjugate gradient, Part I: Theory. *J. Optim. Theory Appl.* **1964**, *7*, 149–154.
4. Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradient. *Comput. J.* **1964**, *7*, 149–154. [[CrossRef](#)]
5. Dai, Y.H.; Yuan, Y.X. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **2000**, *10*, 177–182. [[CrossRef](#)]
6. Fletcher, R. *Practical Methods of Optimization, Unconstrained Optimization*; Wiley: New York, NY, USA, 1988; Volume I.
7. Li, Y.F.; Liu, Z.X.; Liu, H.W. A subspace minimization conjugate gradient method based on conic model for unconstrained optimization. *Comput. Appl. Math.* **2019**, *38*, 16. [[CrossRef](#)]
8. Dai, Y.H.; Kou, C.X. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **2013**, *23*, 296–320. [[CrossRef](#)]
9. Wang, T.; Liu, Z.X.; Liu, H.W.; Wang, T.; Liu, Z.; Liu, H. A new subspace minimization conjugate gradient method based on tensor model for unconstrained optimization. *Int. J. Comput. Math.* **2019**, *96*, 1924–1942. [[CrossRef](#)]
10. Li, M.; Liu, H.W.; Liu, Z.X. A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization. *Numer. Algorithms* **2018**, *79*, 195–219. [[CrossRef](#)]
11. Liu, H.; Liu, Z. An Efficient Barzilai-Borwein Conjugate Gradient Method for Unconstrained Optimization. *J. Optim. Theory Appl.* **2019**, *180*, 879–906. [[CrossRef](#)]
12. Yuan, Y.; Stoer, J. A subspace study on conjugate gradient algorithms. *Zamm J. Appl. Math. Mech. Z. Angew. Math. Mech.* **1995**, *75*, 69–77. [[CrossRef](#)]
13. Andrei, N. An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. *Numer. Algorithms* **2014**, *65*, 859–874. [[CrossRef](#)]
14. Yang, Y.T.; Chen, Y.T.; Lu, Y.L. A subspace conjugate gradient algorithm for large-scale unconstrained optimization. *Numer. Algorithms* **2017**, *76*, 813–828. [[CrossRef](#)]
15. Dai, Y.H.; Kou, C.X. A Barzilai-Borwein conjugate gradient method. *Sci. China Math.* **2016**, *59*, 1511–1524. [[CrossRef](#)]
16. Barzilai, J.; Borwein, J.M. Two-point step size gradient methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [[CrossRef](#)]
17. Wang, Z.H.; Yuan, Y. A subspace implementation of quasi-Newton trust region methods for unconstrained optimization. *Numer. Math.* **2006**, *104*, 241–269. [[CrossRef](#)]
18. Yuan, Y.X. A review on subspace methods for nonlinear optimization. In Proceedings of the International Congress of Mathematics, Seoul, Korea, 13–21 August 2014; pp. 807–827.
19. Fialko, S.; Karpilovsky, V. Block subspace projection preconditioned conjugate gradient method in modal structural analysis. *Comput. Math. Appl.* **2020**, *79*, 3410–3428. [[CrossRef](#)]
20. Hanzely, F.; Doikov, N.; Nesterov, Y.; Richtarik, P. Stochastic subspace cubic Newton method. In Proceedings of the International Conference on Machine Learning, PMLR, Montreal, QC, Canada, 6–8 July 2020; pp. 4027–4038.
21. Moufawad, S.M. s-Step Enlarged Krylov Subspace Conjugate Gradient Methods. *SIAM J. Sci. Comput.* **2020**, *42*, A187–A219. [[CrossRef](#)]
22. Soodhalter, K.M.; de Sturler, E.; Kilmer, M.E. A survey of subspace recycling iterative methods. *GAMM-Mitteilungen* **2020**, *43*, e202000016. [[CrossRef](#)]
23. Babaie-Kafaki, S. Two modified scaled nonlinear conjugate gradient method. *J. Comput. Appl. Math.* **2014**, *261*, 172–182. [[CrossRef](#)]
24. Yao, S.; Wu, Y.; Yang, J.; Xu, J. A three-term gradient descent method with subspace techniques. *Math. Probl. Eng.* **2021**, *2021*, 8867309. [[CrossRef](#)]
25. Andrei, N. An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. *Numer. Algorithms* **2006**, *42*, 63–73. [[CrossRef](#)]
26. Zhang, H.C.; Hager, W.W. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.* **2004**, *14*, 1043–1105. [[CrossRef](#)]
27. Hager, W.W.; Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **2005**, *16*, 170–192. [[CrossRef](#)]

28. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–203. [[CrossRef](#)]
29. Versaci, M.; Calcagno, S.; Morabito, F.C. Fuzzy geometrical approach based on unit hyper-cubes for image contrast enhancement. In Proceedings of the 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 19–21 October 2015; pp. 488–493.
30. Rahim, S.S.; Palade, V.; Shuttleworth, J.; Jayne, C. Automatic screening and classification of diabetic retinopathy and maculopathy using fuzzy image processing. *Brain Inform.* **2016**, *3*, 249–267. [[CrossRef](#)] [[PubMed](#)]
31. Hanjing, A.; Suantai, S. A fast image restoration algorithm based on a fixed point and optimization method. *Mathematics* **2020**, *8*, 378. [[CrossRef](#)]
32. Padcharoen, A.; Kitkuan, D. Iterative methods for optimization problems and image restoration. *Carpathian J. Math.* **2021**, *37*, 497–512. [[CrossRef](#)]
33. Ibrahim, A.H.; Kumam, P.; Kumam, W. A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration. *IEEE Access* **2020**, *8*, 162714–162729. [[CrossRef](#)]
34. Fessler, J.A. Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms. *IEEE Signal Process. Mag.* **2020**, *37*, 33–40. [[CrossRef](#)]