

## Article

# A Class of Three-Dimensional Subspace Conjugate Gradient Algorithms for Unconstrained Optimization

Jun Huo <sup>1,†</sup>, Jielan Yang <sup>2,\*,†</sup>, Guoxin Wang <sup>2</sup> and Shengwei Yao <sup>1,3,\*</sup>

<sup>1</sup> Guangxi (ASEAN) Financial Research Center, Guangxi University of Finance and Economics, Nanning 530007, China; 2018210016@gxuufe.edu.cn

<sup>2</sup> School of Mathematics and Information Science, Guangxi University, Nanning 530004, China; Wgxin@st.gxu.edu.cn

<sup>3</sup> Guangxi Key Laboratory Cultivation Base of Cross-Border E-Commerce Intelligent Information Processing, Guangxi University of Finance and Economics, Nanning 530007, China

\* Correspondence: yangjl@st.gxu.edu.cn (J.Y.); ysw@gxu.edu.cn (S.Y.)

† These authors contributed equally to this work.

**Abstract:** In this paper, a three-parameter subspace conjugate gradient method is proposed for solving large-scale unconstrained optimization problems. By minimizing the quadratic approximate model of the objective function on a new special three-dimensional subspace, the embedded parameters are determined and the corresponding algorithm is obtained. The global convergence result of a given method for general nonlinear functions is established under mild assumptions. In numerical experiments, the proposed algorithm is compared with SMCG\_NLS and SMCG\_Conic, which shows that the given algorithm is robust and efficient.

**Keywords:** unconstrained optimization; conjugate gradient method; subspace method; quadratic model



**Citation:** Huo, J.; Yang, J.; Wang, G.; Yao, S. A Class of Three-Dimensional Subspace Conjugate Gradient Algorithms for Unconstrained Optimization. *Symmetry* **2022**, *14*, 80. <https://doi.org/10.3390/sym14010080>

Academic Editor: José Carlos R. Alcantud

Received: 19 November 2021

Accepted: 16 December 2021

Published: 5 January 2022

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



**Copyright:** © 2022 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 method is one of the most important methods used for solving large-scale unconstrained problems, because of its simple structure, lower computation, storage, fast convergence, etc. The general unconstrained optimization problem is as follows:

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

where  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differentiable. The function value of  $f(x)$  at  $x_k$  is denoted as  $f_k$ , and its gradient is expressed as  $g_k$ . Let  $\alpha_k$  be the step size; we have the following iteration form for conjugate gradient method:

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

where  $d_k$  is the search direction, which has the form of

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

where  $\beta_k \in \mathbb{R}$ , referred to as the conjugate parameter. For different selections of  $\beta_k$ , there are several well-known nonlinear conjugate gradient methods [1–6]:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

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

where  $\|\cdot\|$  represents the Euclidean norm, and  $y_k = g_{k+1} - g_k$ .

The step size  $\alpha_k$  can be obtained in different ways. Zhang and Hager [7] proposed an effective non-monotone Wolfe line search as follows:

$$f(x_k + \alpha_k d_k) \leq C_k + \delta \alpha_k g_k^T d_k, \quad (4)$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k. \quad (5)$$

The  $C_k$  in the formula (4) is the convex combination of  $f_0, f_1, \dots, f_k$ .  $0 < \delta < \sigma < 1$ ,  $C_0 = f_0$ ,  $Q_0 = 1$ ,  $C_{k+1}$  and  $Q_{k+1}$  are updated by the following rule:

$$C_{k+1} = \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}}, \quad Q_{k+1} = \eta_k Q_k + 1. \quad (6)$$

where  $0 \leq \eta_{\min} \leq \eta_k \leq \eta_{\max} \leq 1$ .

The generalized non-monotone line search in [8] is composed of formula (5) and

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha_k d_k) + \eta_k + \delta \alpha_k g_k^T d_k, \quad (7)$$

where  $0 < \delta < \sigma < 1$ , and

$$\eta_k = \begin{cases} 0, & \text{if } k = 0, \\ \min\left\{\frac{1}{(k \lg(k/n+12))}, C_k - f(x_k)\right\}, & \text{if } k \geq 1. \end{cases} \quad (8)$$

For large-scaled optimization problems, some researchers have been looking for more efficient algorithms. In 1995, Yuan and Stoer [9] first proposed the method of embedding subspace technology into the conjugate gradient algorithm framework, namely the two-dimensional subspace minimization conjugate gradient method (SMCG for short). The search direction is calculated by minimizing the quadratic approximation model on the two-dimensional subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k\}$ , namely

$$d_{k+1} = \mu_k g_{k+1} + \nu_k s_k,$$

where  $\mu_k$  and  $\nu_k$  are parameters, and  $s_k = x_{k+1} - x_k$ . Analogously, the calculation of the search direction is directly extended to  $\text{Span}\{g_{k+1}, s_k, s_{k-1}\}$ . By this way, we can avoid solving the sub-problem in the total space, which can reduce the computation and storage cost immensely.

Inspired by SMCG, some researchers began to investigate the algorithm of the conjugate gradient method combined with subspace technology. Dai et al. [10] focused on the analysis of the subspace minimization conjugate gradient method proposed by Yuan and Stoer [9] and integrated SMCG with Barzilai–Borwein [11], a new Barzilai–Borwein conjugate gradient method (BBCG for short) was proposed. In the subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k\}$ , Li et al. [12] discussed the case where the search direction was generated by minimizing the conic model when the non-quadratic state of the objective function was stronger. Wang et al. [13] changed the conic model of [12] into the tensor model. Zhao et al. [14] discussed the case of regularization model. Andrei [15] further expanded the search direction, developed it into a three-dimensional subspace  $\Omega_{k+1} = \text{Span}\{-g_{k+1}, s_k, y_k\}$ , and proposed a new SMCG method (TTS). Inspired by Andrei, Yang et al. [16] carried out a similar study. They applied the subspace minimization technique to another special three-dimensional subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, s_{k-1}\}$ , and obtained a new SMCG method (STT). On the same subspace, Li et al. [17] further studied Yang's results, analyzed the more complex three parameters, and proposed a new subspace minimization conjugate gradient method (SMCG\_NLS). Yao et al. [18] proposed a new three-dimensional subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, g_k\}$  and obtained the TCGS method by using the modified secant equation.

According to [19], it can be seen that the key to embedding subspace technology into the conjugate gradient method is to construct an appropriate subspace, select an approximate model, and estimate the terms of the Hessian matrix. The subspace  $\Omega_{k+1} = \text{Span}\{-g_{k+1}, s_k, y_k\}$  contains the gradient of the two iteration points. If the difference between the two gradient points is too large, the direction of the gradient change  $y_k$  and its value will be affected. We consider making appropriate corrections to the gradient changes of the two iteration points, and then combine the current iteration point gradient with the previous search direction to form a new three-dimensional subspace, and whether this subspace is valuable for research. This paper is taking it as the breakthrough point for research.

In this paper, to avoid the situation in which the changes of the gradient  $y_k = g_{k+1} - g_k$  dominate the subspace  $\text{Span}\{g_{k+1}, d_k, y_k\}$ , inspired by [20], we construct a similar subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, y_k^*\}$  in which  $y_k^* = g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k$ , and then, by solving the optimal solution of an approximate model of the objective function in the given subspace to gain the corresponding parameters and algorithm. It can be shown that the obtained method is a global convergent and has nice numerical performance.

The rest of this paper is organized as follows: in Section 2, the search direction constructed on a new special three-dimensional subspace  $\Omega_{k+1}$  is presented, and the estimations of matrix-vector production are given. In Section 3, The proposed algorithm and its properties under two necessary assumptions are described in detail. In Section 4, we establish the global convergence of our proposed algorithm under mild conditions. In Section 5, we compare the proposed method numerically with algorithms SMCG\_NLS [17] and SMCG\_Conic [12]. Finally, in Section 6, we conclude this paper and highlight future work.

## 2. Search Direction and Step Size

The main content of this section is to introduce four search direction models and the selections of initial step sizes on the newly spanned three-dimensional subspace.

### 2.1. Direction Choice Model

Inspired by [20], in this paper, the gradient change  $y_k = g_{k+1} - g_k$  is replaced by  $y_k^* = g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k$ . Then, the search directions are constructed in the three-dimensional subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, y_k^*\}$ .

From [10], we know that the approximate model used plays two roles: one is to approximate the original objective function in the subspace  $\Omega_{k+1}$ ; the other one is to make the search direction  $d_{k+1}$  obtained by the approximate model descend, so that the original objective function declines along this direction. On our proposed subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, y_k^*\}$ , we consider the approximate model of the objective function as

$$\min_{d \in \Omega_{k+1}} \phi_{k+1}(d) = g_{k+1}^T d + \frac{1}{2} d^T B_{k+1} d, \quad (9)$$

where  $B_{k+1}$  is a symmetric positive definite approximation matrix of the Hessian matrix, satisfying  $B_{k+1} s_k = y_k$ .

Obviously, there are three dimensions in the subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, y_k^*\}$  that we are considering.

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

When the dimension is 3, it is easy to know that  $g_{k+1}$ ,  $s_k$ ,  $y_k^*$  are not collinear. The form of the search direction is of the following form

$$d_{k+1} = \mu_k g_{k+1} + \nu_k s_k + \gamma_k y_k^*, \quad (10)$$

where  $\mu_k$ ,  $\nu_k$  and  $\gamma_k$  are undetermined parameters. Substituting (10) into (9) and simplifying, we have

$$\min_{(\mu, \nu, \gamma)} \left\{ \left( \begin{array}{c} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{array} \right)^T \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right) + \frac{1}{2} \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right)^T \left( \begin{array}{ccc} \rho_{k+1} & g_{k+1}^T y_k & w_k \\ g_{k+1}^T y_k & s_k^T y_k & y_k^T y_k^* \\ w_k & y_k^T y_k^* & q_k \end{array} \right) \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right) \right\}, \quad (11)$$

where  $\rho_{k+1} = g_{k+1}^T B_{k+1} g_{k+1}$ ,  $q_k = g_{k+1}^T B_{k+1} y_k^*$ ,  $w_k = (y_k^*)^T B_{k+1} y_k^*$ . Set

$$D_{k+1} = \left( \begin{array}{ccc} \rho_{k+1} & g_{k+1}^T y_k & w_k \\ g_{k+1}^T y_k & s_k^T y_k & y_k^T y_k^* \\ w_k & y_k^T y_k^* & q_k \end{array} \right).$$

Thus, (11) can be summarized as

$$\min_{(\mu, \nu, \gamma)} \left\{ \left( \begin{array}{c} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{array} \right)^T \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right) + \frac{1}{2} \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right)^T D_{k+1} \left( \begin{array}{c} \mu \\ \nu \\ \gamma \end{array} \right) \right\}.$$

Under some mild conditions, we can prove that  $D_{k+1}$  is positive definite, which will be discussed in Lemma 1. When  $D_{k+1}$  is positive definite, by calculation and simplification, the only solution of (11) is

$$\left( \begin{array}{c} \mu_k \\ \nu_k \\ \gamma_k \end{array} \right) = -\frac{1}{\Delta_{k+1}} \left( \begin{array}{ccc} \chi & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & \omega \end{array} \right) \left( \begin{array}{c} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{array} \right), \quad (12)$$

where

$$\begin{aligned} \Delta_{k+1} &= |D_{k+1}| = \rho_{k+1}\chi + w_k\theta_2 + g_{k+1}^T y_k\theta_1, \\ \chi &= q_k(s_k^T y_k) - (y_k^T y_k^*)^2, \\ \omega &= \rho_{k+1}(s_k^T y_k) - (g_{k+1}^T y_k)^2, \\ \theta_1 &= w_k(y_k^T y_k^*) - q_k(g_{k+1}^T y_k), \\ \theta &= q_k\rho_{k+1} - w_k^2, \\ \theta_2 &= (y_k^T y_k^*)(g_{k+1}^T y_k) - w_k(s_k^T y_k), \\ \theta_3 &= w_k(g_{k+1}^T y_k) - \rho_{k+1}(y_k^T y_k^*). \end{aligned}$$

In order to avoid the matrix-vector multiplication, we need to estimate  $\rho_{k+1}$ ,  $q_k$  and  $w_k$ . Before estimating  $\rho_{k+1}$ , we first estimate  $q_k$ ,  $w_k$ .

For  $q_k$ , we get

$$\begin{aligned} q_k &= (y_k^*)^T B_{k+1} y_k^* \\ &= \left( \frac{(y_k^*)^T B_{k+1} y_k^* s_k^T B_{k+1} s_k}{((y_k^*)^T B_{k+1} s_k)^2} \right) \left( \frac{((y_k^*)^T B_{k+1} s_k)^2}{s_k^T B_{k+1} s_k} \right) \\ &= \frac{\left\| B_{k+1}^{\frac{1}{2}} y_k \right\|^2 \left\| B_{k+1}^{\frac{1}{2}} s_k \right\|^2 (y_k^T y_k^*)^2}{\left( \left( B_{k+1}^{\frac{1}{2}} y_k^* \right)^T \left( B_{k+1}^{\frac{1}{2}} s_k \right) \right)^2 s_k^T y_k} \\ &= \frac{1}{\cos^2 \left\langle B_{k+1}^{\frac{1}{2}} y_k^*, B_{k+1}^{\frac{1}{2}} s_k \right\rangle} \frac{(y_k^T y_k^*)^2}{s_k^T y_k}. \end{aligned}$$

Based on the analysis of [9],  $\cos^2 \left\langle B_{k+1}^{\frac{1}{2}} y_k^*, B_{k+1}^{\frac{1}{2}} s_k \right\rangle$  is desirable, which shows that  $q_k$  can have the following estimation:

$$q_k = 2 \frac{(y_k^T y_k^*)^2}{s_k^T y_k}. \quad (13)$$

which also means that

$$q_k s_k^T y_k - (y_k^T y_k^*)^2 = (y_k^T y_k^*)^2.$$

In order for the experiment to have a better numerical effect, we amended  $q_k$  as

$$q_k = \frac{(y_k^T y_k^*)^2}{s_k^T y_k} + \lambda_k, \quad (14)$$

where

$$\lambda_k = \max \left\{ \frac{(y_k^T y_k^*)^2}{s_k^T y_k}, 0.1 \|y_k^*\|^2 \right\}.$$

Obviously,  $q_0 > 0$ . Consider the following matrix,

$$\begin{pmatrix} s_k^T y_k & y_k^T y_k^* \\ y_k^T y_k^* & q_k \end{pmatrix}. \quad (15)$$

From (13) and  $q_0 > 0$ , it can be known that the sub-matrix (15) is positive definite, inspired by the BBCG method [11], for  $w_k$ , we take

$$w_k = \zeta_k \frac{g_{k+1}^T y_k^* \|y_k\|^2}{s_k^T y_k}, \quad (16)$$

where

$$\zeta_k = \begin{cases} \max\{0.9\zeta_{k-1}, 1.2\}, & \text{if } \alpha_k > 1, \\ \min\{1.1\zeta_{k-1}, 1.75\}, & \text{otherwise,} \end{cases} \quad (17)$$

where  $\zeta_0 = 1.5$ ,  $\zeta_k \in [1.2, 1.75]$ .

Now, we estimate  $\rho_{k+1}$ . Since  $D_{k+1}$  is positive definite, it is easy to know  $\Delta_{k+1} > 0$ , therefore

$$\rho_{k+1} \geq \frac{-w_k \theta_2 - g_{k+1}^T y_k \theta_1}{\chi}. \quad (18)$$

Note that the right-hand side of (18) is  $n_k$ . Combining with (12), (14) and (16), we have

$$n_k = \frac{1}{m_k} \left( \frac{w_k^2}{q_k} + \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k} - \frac{2w_k (g_{k+1}^T y_k) (y_k^T y_k^*)}{q_k (s_k^T y_k)} \right), \text{ with } m_k = 1 - \frac{(y_k^T y_k^*)^2}{q_k (s_k^T y_k)}. \quad (19)$$

According to (14), we know that  $m_k \geq 1/2$ .

In order to ensure that (18) holds, we estimate the parameter  $\rho_{k+1}$  by taking

$$\rho_{k+1} = \zeta_k \max\{n_k, K\}, \quad (20)$$

where  $K = K_1 \|g_{k+1}\|^2$ ,  $K_1 = \max \left\{ \frac{\|y_k\|^2}{s_k^T y_k}, \frac{4 \|y_k\|^4 \|y_k^*\|^2}{q_k (s_k^T y_k)^2} \right\}$ , and  $\rho_0 \in (0, 1)$ . Through the debugging of the algorithm, we find that the numerical experiment effect of  $\zeta_k$  using the adaptive value of (17) is better than that of  $\zeta_k$  with a fixed value, so  $\zeta_k$  also uses (17).

In summary, we find that when the following conditions are satisfied, the search direction  $d_{k+1}$  is calculated by (10) and (12).

$$\xi_3 \leq \frac{\|s_k\|^2}{\|g_{k+1}\|^2}, \quad (21)$$

$$\xi_1 \leq \frac{s_k^T y_k}{\|s_k\|^2} \leq \frac{\|y_k\|^2}{s_k^T y_k} \leq \xi_2, \quad (22)$$

$$\xi_1 \leq \frac{q_k}{\|y_k^*\|^2}, \quad \frac{4\|y_k\|^4 \|y_k^*\|^2}{q_k (s_k^T y_k)^2} \leq \xi_2, \quad (23)$$

where  $\xi_1, \xi_2, \xi_3$  are positive constants.

Now, let us prove that  $D_{k+1}$  is positive definite.

**Lemma 1.** Assuming that the conditions (21)–(23) hold, and  $\rho_{k+1}$  is calculated by (20), the matrix  $D_{k+1}$  in (10) is positive definite.

**Proof of Lemma 1.** Using mathematical induction, it is easy to know  $\rho_0 \in (0, 1)$ ,  $\rho_{k+1} \geq \xi_k K > 0$  from (18); for  $\rho_{k+1} > \frac{\|g_{k+1}\|^2 \|y_k\|^2}{s_k^T y_k}$ , therefore

$$\Delta_{k+1} = |D_{k+1}| = \rho_{k+1} \chi + w_k \theta_2 + g_{k+1}^T y_k \theta_1 > 0.$$

So the proof is over.  $\square$

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

In this case, the form of the search direction  $d_{k+1}$  is as follows:

$$d_{k+1} = \mu_k g_{k+1} + \nu_k s_k, \quad (24)$$

where  $\mu_k, \nu_k$  are undetermined parameters. Substituting (24) into (9), we get

$$\min_{(\mu, \nu)} \left\{ \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \end{pmatrix}^T \begin{pmatrix} \mu \\ \nu \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mu \\ \nu \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k \\ g_{k+1}^T y_k & s_k^T y_k \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\}. \quad (25)$$

where  $\rho_{k+1} = g_{k+1}^T B_{k+1} g_{k+1}$ , if it satisfies

$$\Delta_{k+1} = \rho_{k+1} (s_k^T y_k) - (g_{k+1}^T y_k)^2 > 0.$$

Then the unique solution of the problem (25) is

$$\begin{pmatrix} \mu_k \\ \nu_k \end{pmatrix} = \frac{1}{\Delta_{k+1}} \begin{pmatrix} (g_{k+1}^T y_k)(g_{k+1}^T s_k) - (s_k^T y_k) \|g_{k+1}\|^2 \\ (g_{k+1}^T y_k) \|g_{k+1}\|^2 - \rho_{k+1} (g_{k+1}^T s_k) \end{pmatrix}. \quad (26)$$

Combined with BBFG, there is

$$\rho_{k+1} = \xi_k \frac{\|g_{k+1}\|^2 \|y_k\|^2}{s_k^T y_k}, \quad (27)$$

where  $\xi_k$  in (27) is the same as (17).

Obviously, under certain conditions, the HS direction can be regarded as a special case of formula (24). Taking into account the finite termination of the HS method, in order to make our algorithm have good properties, when the following conditions hold,

$$\xi_1 \leq \frac{s_k^T y_k}{\|s_k\|^2}, \quad (28)$$

$$\frac{|g_{k+1}^T y_k g_{k+1}^T d_k|}{d_k^T y_k \|g_{k+1}\|^2} \leq \xi_4, \quad (29)$$

where  $\xi_4 \in [0, 1)$ . We consider

$$d_{k+1} = -g_{k+1} + \beta_k d_k. \quad (30)$$

In summary, for the case where the dimension of the subspace is 2; if only condition (22) is true,  $d_{k+1}$  is calculated by (24). When inequalities (28) and (29) hold,  $d_{k+1}$  is calculated from (30).

According to the above analysis,  $d_{k+1}$  is calculated by (24) when only condition (22) is true for the case of the 2-subspace dimension. When conditions (28) and (29) hold,  $d_{k+1}$  is calculated by (30).

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

When all conditions (21), (22), and (23) are not valid, we choose the negative gradient as the search direction, i.e.,

$$d_{k+1} = -g_{k+1}. \quad (31)$$

## 2.2. Selection of Initial Step Size

Considering that the selections of initial step sizes will also have an impact on the algorithm, we choose the initial step size selection method of SMCG-NLS [17], which is also a subspace algorithm.

According to [21], we know that

$$t_k = \left| \frac{2(f_k - f_{k+1} + g_{k+1}^T s_k)}{s_k^T y_k} - 1 \right|, \quad (32)$$

which indicates how close the objective function is to the quadratic function on the line segment formed between the current iteration point and the previous iteration point. Based on [22], we know that the following condition indicates that the objective function is close to a quadratic function:

$$t_k \leq \lambda_1 \text{ or } \max\{t_k, t_{k-1}\} \leq \lambda_2, \quad (33)$$

where  $0 < \lambda_1 < \lambda_2$ .

**Case I:** when the search direction is calculated by Equations (10), or (24) or (30), the initial step size is  $\alpha_k^{(0)} = 1$ .

**Case II:** when  $d_{k+1} = -g_{k+1}$ , the initial step size is

$$\tilde{\alpha}_k = \begin{cases} \max\{\min\{\alpha_k^{(BB2)}, \vartheta_{\max}\}, \vartheta_{\min}\}, & \text{if } g_{k+1}^T s_k \geq 0, \\ \max\{\min\{\alpha_k^{(BB1)}, \vartheta_{\max}\}, \vartheta_{\min}\}, & \text{if } g_{k+1}^T s_k < 0, \end{cases} \quad (34)$$

where  $\alpha_k^{(BB1)} = \frac{\|s_k\|^2}{s_k^T y_k}$ ,  $\alpha_k^{(BB2)} = \frac{s_k^T y_k}{\|y_k\|^2}$ .

## 3. The Obtained Algorithm and Descent Property

This section describes the obtained algorithm and its descending properties under two necessary assumptions in detail.

### 3.1. The Obtained Algorithm

The main content of this section is to introduce our proposed algorithm and give two necessary assumptions. Before introducing the algorithm, we first introduce the restart method we use to restart the proposed algorithm, and then describe the proposed algorithm in detail.

According to [23], set

$$r_k = \frac{2(f_{k+1} - f_k)}{g_{k+1}^T s_k + g_k^T s_k},$$

If  $r_k$  is close to 1, then the one-dimensional line search function is close to the quadratic function. Similar to [21], if in multiple consecutive iterations,  $|r_k - 1| \leq \xi_5$ , we restart the search direction along  $-g_{k+1}$ . In addition, we restart our algorithm if the number of consecutive uses of CG directions reaches the MaxRestart threshold.

Now, the details of the three-term subspace conjugate gradient method (TSCG for short) is given as follows:

---

**Algorithm 1:** TSCG Alogrithm

---

1. Given  $x_0 \in \mathbb{R}^n$ ,  $\alpha_0^{(0)}$ ,  $\varepsilon > 0$ ,  $0 < \delta < \sigma < 1$ ,  $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \lambda_1, \lambda_2 \in [0, 1)$ , MaxRestart, MinQuad. Let  $C_0 = f_0$ ,  $Q_0 = 1$ , IterRestart := 0, Numgrad := 0, IterQuad := 0, Numcongrad := 0, and  $k := 0$ .
  2. When  $\|g_k\| \leq \varepsilon$ , stop; otherwise, let  $d_0 := -g_0$ , Numgrad = 1.
  3. If  $k = 0$ , go to step 4. If  $d_{k+1} = -g_{k+1}$ , via (34) Compute  $\alpha_k^{(0)}$ ; otherwise,  $\alpha_k^{(0)} = 1$ .
  4. Calculate the step size  $\alpha_k$  by (7) and (5).
  5. Update  $x_{k+1}$  with (2). When  $\|g_k\| \leq \varepsilon$ , stop; otherwise, let  $\text{IterRestart} = \text{IterRestart} + 1$ . If  $|r_k - 1| \leq \xi_5$  or  $|f_{k+1} - f_k - (g_{k+1}^T s_k + g_k^T s_k)| \leq \xi_6/6$ ,  $\text{IterQuad} = \text{IterQuad} + 1$ ; otherwise,  $\text{IterQuad} = 0$ .
  6. Calculate the search direction  $d_{k+1}$ .
    - (a) If the conditions (21), (22) and (23) hold, calculate  $d_{k+1}$  by (10) and (12), and then perform step 7;
    - (b) If the condition (22) holds, calculate  $d_{k+1}$  by (24) and (26), and then perform step 7;
    - (c) If the condition (28) and (29) hold, calculate  $d_{k+1}$  by (30), and then perform step 7;
    - (d) Otherwise, calculate  $d_{k+1}$  from (31), and then perform step 7.
  7. Update  $Q_{k+1}$  and  $C_{k+1}$  by (6).
  8. Calculate  $\eta_{k+1}$  by (8), and set  $k := k + 1$ , and go to step 1.
- 

### 3.2. Descent Properties of Search Direction

In this subsection, we discuss the descent properties of the given algorithm, in which it will be proved that the proposed algorithm (TSCG) fulfills sufficient descent conditions in all cases. Now, we first introduce some common assumptions on the objective function.

**Assumption 1.** The objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuous differentiable and has a lower bound on  $\mathbb{R}^n$ .

**Assumption 2.** The gradient function  $g$  is Lipschitz continuous on the bounded level set  $\Theta = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ ; that is, there exists constant  $L > 0$ , such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Theta.$$

That is:  $\|y_k\| \leq L\|s_k\|$ .

**Lemma 2.** If the search direction  $d_{k+1}$  is generated by (10) or (24), then

$$g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}, \quad (35)$$

holds.

**Proof of Lemma 2.** We only need to discuss the situation of search direction in relation to  $\rho_{k+1}$ .

**Case I:** if  $d_{k+1}$  is generated by (24), the proof is similar to [22], so the proof process is omitted.

**Case II:** when  $d_{k+1}$  is calculated by (10) and (12), we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{pmatrix}^T \begin{pmatrix} \mu_k \\ \nu_k \\ \gamma_k \end{pmatrix} \\ &= -\frac{1}{\Delta_{k+1}} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{pmatrix}^T \begin{pmatrix} \chi & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & \omega \end{pmatrix} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k^* \end{pmatrix} \\ &= -\frac{\|g_{k+1}\|^4}{\Delta_{k+1}} \varphi(x, y), \end{aligned}$$

where  $\varphi(x, y)$  is the binary quadratic function of variables  $\frac{g_{k+1}^T y_k^*}{\|g_{k+1}\|^2}$  and  $\frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}$  represented as  $x$  and  $y$ , and its simplified form is expressed as

$$\varphi(x, y) = \omega x^2 + 2\theta_3 xy + \theta y^2 + 2\theta_2 x + 2\theta_1 y + \chi.$$

From Lemma 1, it is easy to get  $\omega > 0$ ,  $\omega\theta - \theta_3^2 = \Delta_{k+1}\rho_{k+1} > 0$ ; that is

$$\varphi(x, y)_{\min} = \frac{\Delta_{k+1}}{\rho_{k+1}}.$$

Therefore, we have

$$g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\Delta_{k+1}} \varphi(x, y)_{\min} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}.$$

Thus, that is the end of the proof.  $\square$

**Lemma 3.** Assume that  $d_{k+1}$  is generated by the algorithm TSCG. there exists a constant  $c_1 > 0$ , such that

$$g_{k+1}^T d_{k+1} \leq -c_1 \|g_{k+1}\|^2. \quad (36)$$

**Proof of Lemma 3.** As for the four forms of the constituent directions, we discuss them separately.

**Case I:** if  $d_{k+1} = -g_{k+1}$ , let  $c_1 = \frac{1}{2}$ , then (36) holds.

**Case II:** when the search direction is binomial, namely, we first discuss the situation given by (30). When  $d_{k+1}$  is determined by (30), combined with (28) and (29), for  $\beta_k = \beta_k^{HS}$ , we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \\ &\leq -\|g_{k+1}\|^2 + \frac{|(g_{k+1}^T y_k)(g_{k+1}^T d_k)|}{d_k^T y_k} \\ &\leq -\|g_{k+1}\|^2 + \zeta_4 \|g_{k+1}\|^2 \\ &\leq -(1 - \zeta_4) \|g_{k+1}\|^2. \end{aligned}$$

**Case III:** now, we discuss another case where the search direction is a binomial; that is, the situation where the search direction is generated by (24). Combining (22) and (27), obviously

$$\rho_{k+1} \leq \zeta_k \frac{\|g_{k+1}\|^2 \|y_k\|^2}{s_k^T y_k} \leq 2\zeta_2 \|g_{k+1}\|^2.$$

In combination with the above formula and (35), it can be obtained

$$g_{k+1}^T d_{k+1} \leq -\frac{1}{2\bar{\zeta}_2} \|g_{k+1}\|^2.$$

**Case IV:** when the search direction is trinomial; that is,  $d_{k+1}$  is given by (10) and (12). Consider Lemma 2 and use (20), (21), and  $\chi > 0$ , we first prove that  $\rho_{k+1}$  has an upper bound:

$$\begin{aligned} |n_k| &= \left| \frac{\left( \frac{w_k^2}{q_k} + \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k} - \frac{2w_k(g_{k+1}^T y_k)(y_k^T y_k^*)}{q_k(s_k^T y_k)} \right)}{m_k} \right| \\ &\leq \frac{\left( \frac{4\|y_k\|^4 \|y_k^*\|^2}{q_k(s_k^T y_k)^2} + \frac{\|y_k\|^2}{s_k^T y_k} \right) \|g_{k+1}\|^2 + 2 \frac{w_k}{\sqrt{q_k}} \frac{g_{k+1}^T y_k}{\sqrt{s_k^T y_k}} \frac{y_k^T y_k^*}{\sqrt{q_k s_k^T y_k}}}{m_k} \\ &\leq \frac{\left( 2K_1 \|g_{k+1}\|^2 + 2 \frac{|w_k|}{\sqrt{q_k}} \frac{|g_{k+1}^T y_k|}{\sqrt{s_k^T y_k}} \right)}{m_k} \\ &\leq \frac{(2K + 2\sqrt{K}\sqrt{K})}{m_k} \\ &\leq 8K. \end{aligned}$$

The  $\rho_k$  in the root of the above second inequality is  $q_k$ .

Combining the conditions (22) and (23), it is easy to know that  $K_1$  in (20) has an upper bound, namely

$$K_1 = \max \left\{ \frac{\|y_k\|^2}{s_k^T y_k}, 4 \frac{\|y_k\|^4 \|y_k^*\|^2}{q_k(s_k^T y_k)^2} \right\} \leq \bar{\zeta}_2.$$

There is

$$\rho_{k+1} = \zeta_k \max\{n_k, K\} \leq 16K = 16K_1 \|g_{k+1}\|^2 \leq 16\bar{\zeta}_2 \|g_{k+1}\|^2.$$

Finally, according to Lemma 2, we have

$$g_{k+1}^T d_{k+1} \leq -\frac{\|g_{k+1}\|^4}{\rho_{k+1}} \leq -\frac{1}{16\bar{\zeta}_2} \|g_{k+1}\|^2.$$

In summary, the value of  $c_1$ , satisfying (36), is

$$c_1 = \min \left\{ \frac{1}{2}, \frac{1}{2\bar{\zeta}_2}, (1 - \bar{\zeta}_4), \frac{1}{16\bar{\zeta}_2} \right\}.$$

Then the proof is complete.  $\square$

**Lemma 4.** If the search direction  $d_{k+1}$  is calculated by TSCG, then there exists a constant  $c_2 > 0$ , such that

$$\|d_{k+1}\| \leq c_2 \|g_{k+1}\|. \quad (37)$$

**Proof of Lemma 4.** Similar to Lemma 3, we discuss four cases of search directions, respectively.

**Case I:** if  $d_{k+1} = -g_{k+1}$ , let  $c_2 = 1$ , then (37) is true.

**Case II:** if  $d_{k+1}$  is calculated by (30), according to Assumption 2 and the condition (28), we have

$$\begin{aligned}
\|d_{k+1}\| &= \|-g_{k+1} + \beta_k^{HS} d_k\| \\
&\leq \|g_{k+1}\| + \frac{\|g_{k+1}\| \|y_k\| \|d_k\|}{d_k^T y_k} \\
&\leq \left(1 + \frac{L}{\xi_1}\right) \|g_{k+1}\|.
\end{aligned}$$

**Case III:** if  $d_{k+1}$  is calculated by (24) and (26), then, combining (22) and (27), and the Cauchy inequality, we deduce

$$\begin{aligned}
\Delta_{k+1} &= \rho_{k+1}(s_k^T y_k) - (g_{k+1}^T y_k)^2 \\
&\geq \xi_1 \|s_k\|^2 \left( \rho_{k+1} - \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k} \right) \\
&\geq \frac{1}{5} \xi_1 \|s_k\|^2 \frac{\|g_{k+1}\|^2 \|y_k\|^2}{s_k^T y_k}.
\end{aligned} \tag{38}$$

Combining the above equation, (27), the Cauchy inequality, and the triangle inequality, we can get

$$\begin{aligned}
\|d_{k+1}\| &= \|\mu_k g_{k+1} + \nu_k s_k\| \\
&\leq \frac{1}{\Delta_{k+1}} (|g_{k+1}^T y_k| |g_{k+1}^T s_k| \|g_{k+1}\| + |g_{k+1}^T y_k| \|g_{k+1}\|^2 + |\rho_{k+1}(g_{k+1}^T s_k)| \|s_k\|) \\
&\leq \frac{1}{\Delta_{k+1}} \left( 2\|s_k\| \|y_k\| + \frac{\rho_{k+1} \|s_k\|^2}{\|g_{k+1}\|^2} \right) \|g_{k+1}\|^3 \\
&\leq \frac{5s_k^T y_k}{\xi_1 \|s_k\|^2 \|y_k\|^2} \left( 2\|s_k\| \|y_k\| + \frac{\rho_{k+1} \|s_k\|^2}{\|g_{k+1}\|^2} \right) \|g_{k+1}\| \\
&\leq \left( \frac{10s_k^T y_k}{\xi_1 \|s_k\| \|y_k\|} + \frac{10}{\xi_1} \right) \|g_{k+1}\| \\
&\leq \frac{20}{\xi_1} \|g_{k+1}\|.
\end{aligned}$$

**Case IV:** when the search direction is three; that is, the search direction is calculated by (10) and (12). Similar to Case III, let us first derive the lower bound of  $\Delta_{k+1}$ . According to (16), (18) and (20), we have

$$\begin{aligned}
\Delta_{k+1} &= \rho_{k+1} \chi + w_k \theta_2 + g_{k+1}^T y_k \theta_1 \\
&= \chi \left( \rho_{k+1} - \frac{-w_k \theta_2 - g_{k+1}^T y_k \theta_1}{\chi} \right) \\
&= \chi(\rho_{k+1} - n_k) \\
&\geq \chi(1.2 \max\{n_k, K\} - n_k) \\
&\geq \frac{1}{5} \chi K.
\end{aligned}$$

Let us write  $\chi_1 = \rho_k(s_k^T y_k)$ , and combine that with  $m_k \geq \frac{1}{2}$ , we have  $\chi = \chi_1 m_k \geq \frac{1}{2} \chi_1 > 0$ , therefore

$$\Delta_{k+1} \geq \frac{1}{10} \chi_1 K.$$

Then

$$\begin{aligned}
 \|d_{k+1}\| &= \|\mu_k g_{k+1} + \nu_k s_k + \gamma_k y_k^*\| \\
 &= \frac{1}{\Delta_{k+1}} \begin{pmatrix} \|g_{k+1}\|^2 \\ |g_{k+1}^T s_k| \\ |g_{k+1}^T y_k^*| \end{pmatrix}^T \begin{pmatrix} |\chi| & |\theta_1| & |\theta_2| \\ |\theta_1| & |\theta| & |\theta_3| \\ |\theta_2| & |\theta_3| & |\omega| \end{pmatrix} \begin{pmatrix} \|g_{k+1}\| \\ \|s_k\| \\ \|y_k^*\| \end{pmatrix} \\
 &\leq \frac{\|g_{k+1}\|}{\Delta_{k+1}} (\chi_1 \|g_{k+1}\|^2 + 4a_k \sqrt{\chi_1 K_1} \|g_{k+1}\|^2 + 2b_k \sqrt{\chi_1} (K + \rho_{k+1}) + c_k \rho_{k+1}) \\
 &\leq \frac{10\|g_{k+1}\|}{\chi_1 K} (\chi_1 \|g_{k+1}\|^2 + 4a_k \sqrt{\chi_1 K_1} \|g_{k+1}\|^2 + 2b_k \sqrt{\chi_1} (K + \rho_{k+1}) + c_k \rho_{k+1}) \\
 &= \|g_{k+1}\| \left( \frac{10}{K_1} + \frac{40}{\sqrt{K_1}} \frac{a_k}{\sqrt{\chi_1}} + 340 \times \frac{b_k}{\sqrt{\chi_1}} + 160 \times \frac{c_k}{\chi_1} \right),
 \end{aligned}$$

where  $a_k = \sqrt{\varrho_k} \|s_k\| + \sqrt{s_k^T y_k} \|y_k^*\|$ ,  $b_k = \|s_k\| \|y_k^*\|$ ,  $c_k = \varrho_k \|s_k\|^2 + s_k^T y_k \|y_k^*\|^2$ . From (20), (22) and (23), through calculation and simplification, we obtain

$$\begin{aligned}
 \xi_1 &\leq \frac{\|y_k\|^2}{s_k^T y_k} \leq K_1, \\
 \frac{a_k}{\sqrt{\chi_1}} &\leq \frac{2}{\sqrt{\xi_1}}, \quad \frac{b_k}{\sqrt{\chi_1}} \leq \frac{1}{\xi_1}, \quad \frac{c_k}{\chi_1} \leq \frac{2}{\xi_1}.
 \end{aligned}$$

According to the above results, it can be further deduced

$$\|d_{k+1}\| \leq \frac{750}{\xi_1} \|g_{k+1}\|.$$

According to the four cases of the above analyses, the value of  $c_2$  that satisfies (38) is

$$c_2 = \max \left\{ 1, 1 + \frac{L}{\xi_1}, \frac{20}{\xi_1}, \frac{750}{\xi_1} \right\},$$

The proof is ended.  $\square$

#### 4. Convergence Analysis

The global convergence of the algorithm for general functions is proven in this section.

**Lemma 5.** Suppose  $\alpha_k$  is generated by line search (7) and (5), and satisfies Assumption 2, then

$$\alpha_k \geq \frac{(1-\sigma)|g_k^T d_k|}{L\|d_k\|^2}. \quad (39)$$

**Proof of Lemma 5.** By line search conditions (7) and (5), we have

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

We notice that  $\sigma < 1$ ,  $g_k^T d_k < 0$ , and (39) holds immediately.  $\square$

**Theorem 1.** Suppose Assumption 1 and Assumption 2 are satisfied, and sequence  $x_k$  is generated by algorithm TSCG, then

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (40)$$

**Proof of Theorem 1.** According to (5) and (39), there is

$$\begin{aligned}\delta\alpha_k g_k^T d_k &= -\frac{(1-\sigma)\delta}{L} \left( \frac{g_k^T d_k}{\|d_k\|} \right)^2 \\ &\leq -\frac{(1-\sigma)\delta c_3^2}{Lc_4^2} \|g_k\|^2.\end{aligned}\quad (41)$$

Combining (41) and (7), we get

$$-\frac{(1-\sigma)\delta c_3^2}{Lc_4^2} \|g_k\|^2 \leq f_k + \eta_k - f_{k+1}. \quad (42)$$

Clearly

$$f_k + \eta_k - f_{k+1} \geq 0. \quad (43)$$

The above equation with  $\lim_{k \rightarrow \infty} \eta_k = 0$  is equivalent to

$$\liminf_{k \rightarrow \infty} (f_k - f_{k+1}) \geq \liminf_{k \rightarrow \infty} (f_k + \eta_k - f_{k+1}) + \liminf_{k \rightarrow \infty} (-\eta_k) = l \geq 0.$$

Suppose  $l > 0$ , for  $\varepsilon = \min\left\{1, \frac{l}{2}\right\} > 0$ , for any  $k > N$ , there exists  $N > 0$ , then

$$f_{k+1} < f_k - \frac{\varepsilon}{k}. \quad (44)$$

Combining with (44) and  $\sum_{k=N+1}^{+\infty} \frac{1}{k} = +\infty$ , then  $\lim_{k \rightarrow \infty} f_k = -\infty$ . This contradicts Assumption 1 that  $f_k$  on  $\mathbb{R}^n$  has a lower bound. Therefore,

$$\liminf_{k \rightarrow \infty} (f_k + \eta_k - f_{k+1}) = l = 0.$$

In combination with the above equation, (42) and  $\lim_{k \rightarrow \infty} \eta_k = 0$ , then (40) is true. The proof is complete.  $\square$

## 5. Numerical Results

In this section, we compare the numerical performance of the TSCG algorithm with SMCG\_NLS [17] and SMCG\_Conic [12] algorithms, both of which are subspace minimization algorithms, through numerical experiments to prove the effectiveness of the proposed TSCG algorithm. Performance profiles of Dolan and Moré [24] were used to test the performance of the method. Our test functions were derived from 67 functions in [25], as shown in Table 1. It was programmed and run on a Windows 10 PC with a 1.80-GHz CPU and 16.00 GB memory, 64-bit operating system. We set the termination criteria as:  $\|g_k\| \leq \varepsilon$ , or when the number of iterations of the program exceeded 200,000, and exited when one of them was true. The dimensions of the variables of the test function were 10,000 and 12,000 respectively.

The following shows the selection of parameters and some tags and numerical experiments. The initial step size of the first iteration in this paper uses the adaptive strategy of [26].

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

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

**Table 1.** The test problems.

| No. | Test Problems                               | No. | Test Problems                                   |
|-----|---------------------------------------------|-----|-------------------------------------------------|
| 1   | Extended Freudenstein & Roth function       | 35  | NONDIA function (CUTE)                          |
| 2   | Extended Trigonometric function             | 36  | DQDRTIC function (CUTE)                         |
| 3   | Extended Rosenbrock function                | 37  | EG2 function (CUTE)                             |
| 4   | Generalized Rosenbrock function             | 38  | DIXMAANA - DIXMAANL functions                   |
| 5   | Extended White & Holst function             | 39  | Partial Perturbed Quadratic function            |
| 6   | Extended Beale function                     | 40  | Broyden Tridiagonal function                    |
| 7   | Extended Penalty function                   | 41  | Almost Perturbed Quadratic function             |
| 8   | Perturbed Quadratic function                | 42  | Perturbed Tridiagonal Quadratic function        |
| 9   | Diagonal 1 function                         | 43  | Staircase 1 function                            |
| 10  | Diagonal 3 function                         | 44  | Staircase 2 function                            |
| 11  | Extended Tridiagonal 1 function             | 45  | LIARWHD function (CUTE)                         |
| 12  | Full Hessian FH3 function                   | 46  | POWER function (CUTE)                           |
| 13  | Generalized Tridiagonal 2 function          | 47  | ENGVAL1 function (CUTE)                         |
| 14  | Diagonal 5 function                         | 48  | CRAGGLVY function (CUTE)                        |
| 15  | Extended Himmelblau function                | 49  | EDENSCH function (CUTE)                         |
| 16  | Generalized White & Holst function          | 50  | CUBE function (CUTE)                            |
| 17  | Extended PSC1 function                      | 51  | BDEXP function (CUTE)                           |
| 18  | Extended Powell function                    | 52  | NONSCOMP function (CUTE)                        |
| 19  | Full Hessian FH2 function                   | 53  | VARDIM function (CUTE)                          |
| 20  | Extended Maratos function                   | 54  | SINQUAD function (CUTE)                         |
| 21  | Extended Cliff function                     | 55  | Extended DENSCHNB function (CUTE)               |
| 22  | Perturbed quadratic diagonal function 2     | 56  | Extended DENSCHNF function (CUTE)               |
| 23  | Extended Wood function                      | 57  | LIARWHD function (CUTE)                         |
| 24  | Extended Hiebert function                   | 58  | COSINE function (CUTE)                          |
| 25  | Quadratic QF1 function                      | 59  | SINE function:83                                |
| 26  | Extended quadratic penalty QP1 function     | 60  | Generalized Quartic function                    |
| 27  | Extended quadratic penalty QP2 function     | 61  | Diagonal 7 function                             |
| 28  | Quadratic QF2 function                      | 62  | Diagonal 8 function                             |
| 29  | Extended quadratic exponential EP1 function | 63  | Extended TET function:(Three exponential terms) |
| 30  | Extended Tridiagonal 2 function             | 64  | SINCOS function                                 |
| 31  | FLETCHCR function (CUTE)                    | 65  | Diagonal 9 function                             |
| 32  | TRIDIA function (CUTE)                      | 66  | HIMMELBG function (CUTE)                        |
| 33  | ARGLINB function (CUTE)                     | 67  | HIMMELH function (CUTE)                         |
| 34  | ARWHEAD function (CUTE)                     |     |                                                 |

The parameters of SMCG\_NLS and SMCG\_Conic algorithms are selected according to the values selected in the original paper. The parameters of the TSCG algorithm proposed by us are selected as follows.

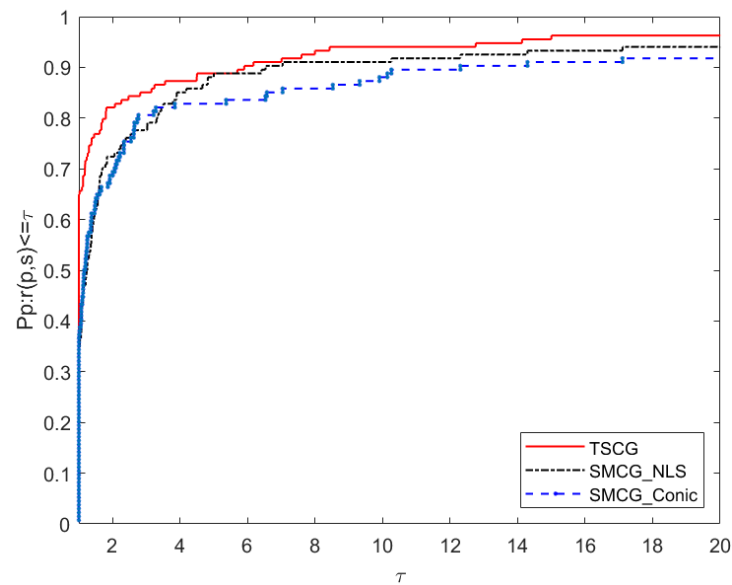
$$\varepsilon = 10^{-6}, \delta = 0.001, \sigma = 0.9999, \xi_1 = 10^{-7}, \xi_2 = 10^6, \xi_3 = 10^{-4}, \xi_4 = 0.875, \\ \xi_5 = 10^{-8}, \xi_6 = 10^{-12}, \lambda_1 = 10^{-5}, \lambda_2 = 0.06, \vartheta_{min} = 10^{-30}, \vartheta_{max} = 10^{30}.$$

The relevant tags of this article are as follows:

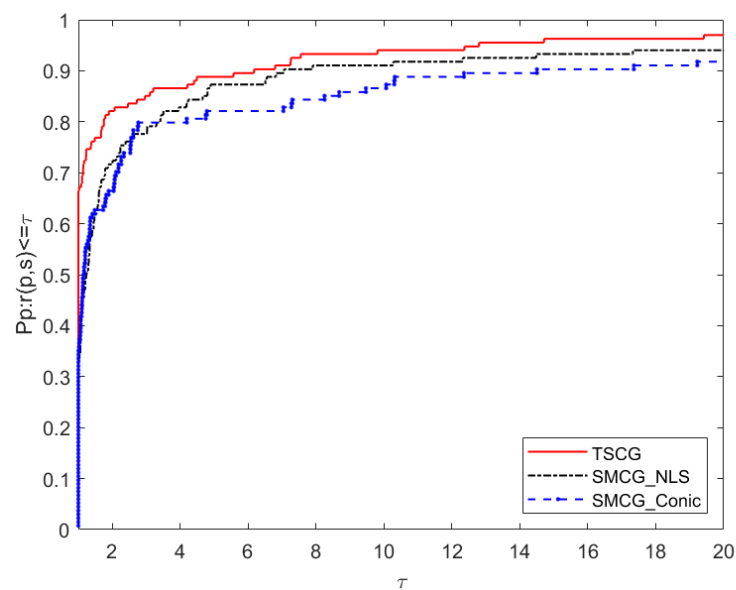
- Ni: the number of iterations.
- Nf: the number of function evaluations.
- Ng: the number of gradient evaluations.
- CPU: the running time of the algorithm (in seconds).

Now, we compare the algorithm TSCG in this paper with SMCG\_NLS [17] and SMCG\_Conic [12] in numerical diagrams. The performance comparison diagrams of Ni, Ng, Nf and CPU correspond to Figures 1–4 respectively. According to Figures 1 and 2, it is found that the robustness and stability of TSCG are significantly better than that of SMCG\_NLS and SMCG\_Conic in terms of the number of iterations and the total number of gradient calculations. Overall, SMCG\_NLS has better robustness and stability than SMCG\_Conic. From Figure 3, we know that both TSCG and SMCG\_NLS are superior to SMCG\_Conic, and TSCG begins to be better than SMCG\_NLS when  $\tau \geq 14.3231$ , even when factor  $\tau \leq 14.3231$ , the total calculated by the function TSCG does not perform as well

as SMCG\_NLS. Figure 4 shows that TSCG and SMCG\_NLS are better than SMCG\_Conic in terms of the robustness and stability of running time. In general, TSCG is almost better than SMCG\_NLS in terms of robustness and stability, only when factor  $2.6328 \leq \tau \leq 7.6472$ , TSCG is inferior to SMCG\_NLS.



**Figure 1.** Performance profiles of the number of iterations ( $N_i$ ).



**Figure 2.** Performance profiles of the gradient ( $N_g$ ).

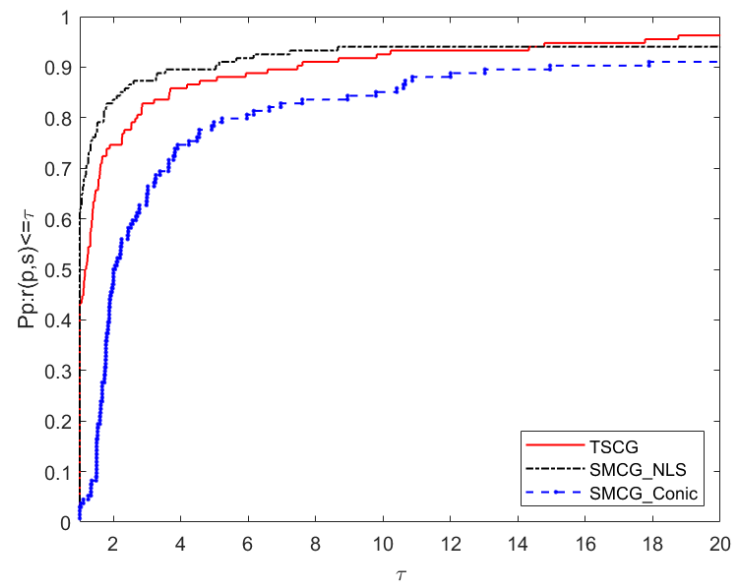


Figure 3. Performance profiles of the function (Nf).

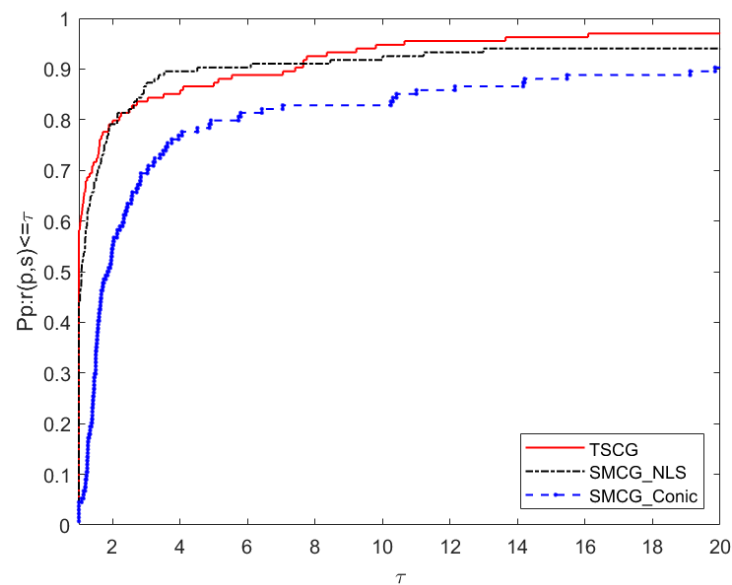


Figure 4. Performance profiles of CPU time (CPU).

## 6. Conclusions and Prospect

In order to obtain a more efficient and robust conjugate gradient algorithm for solving the unconstrained optimization problem, by constructing a new three-dimensional subspace  $\Omega_{k+1} = \text{Span}\{g_{k+1}, s_k, y_k^*\}$ , and solving the sub-problem of a quadratic approximate model of the objective function in the given subspace, we obtained a new three-term conjugate gradient method (TSCG). The global convergence of the TSCG method for general functions is established under mild conditions. Numerical results show that TSCG has better performance than SMCG\_NLS and SMCG\_Conic, both of which are subspace algorithms, for a given test set.

As for the research of the subspace algorithm, the main content of our future work is to continue to study the estimation of terms, including Hessian matrix, to discuss whether it can be extended to the constrained optimization algorithm, and consider the application of the algorithm to image restoration, image segmentation, and path planning of engineering problems.

**Author Contributions:** Conceptualization, J.H. and S.Y.; methodology, J.Y.; software and validation, J.Y. and G.W.; visualization and formal analysis, J.Y.; writing—original draft preparation, J.Y.; supervision, S.Y.; funding acquisition, 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, the Program for the 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

- Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [\[CrossRef\]](#)
- Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149–154. [\[CrossRef\]](#)
- Polyak, B.T. The conjugate gradient method in extremal problems. *JUSSR Comput. Math. Math. Phys.* **1969**, *9*, 94–112. [\[CrossRef\]](#)
- Liu, Y.; Storey, C. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.* **1991**, *69*, 129–137. [\[CrossRef\]](#)
- Dai, Y.H.; Yuan, Y. A nonlinear conjugate gradient method with a strong global convergence property. *Siam J. Optim.* **1999**, *10*, 177–182. [\[CrossRef\]](#)
- Fletcher, R. Volume 1 Unconstrained Optimization. In *Practical Methods of Optimization*; Wiley-Interscience: New York, NY, USA, 1980; p. 120.
- Zhang, H.; Hager, W.W. A non-monotone line search technique and its application to unconstrained optimization. *Soc. Ind. Appl. Math. J. Optim.* **2004**, *14*, 1043–1056.
- Liu, H.; Liu, Z. An efficient Barzilai–Borwein conjugate gradient method for unconstrained optimization. *J. Optim. Theory Appl.* **2019**, *180*, 879–906. [\[CrossRef\]](#)
- Yuan, Y.X.; Stoer, J. A subspace study on conjugate gradient algorithms. *ZAMM-J. Appl. Math. Mech. Für Angew. Math. Und Mech.* **1995**, *75*, 69–77. [\[CrossRef\]](#)
- Dai, Y.H.; Kou, C.X. A Barzilai–Borwein conjugate gradient method. *Sci. China Math.* **2016**, *59*, 1511–1524. [\[CrossRef\]](#)
- Barzilai, J.; Borwein, J.M. Two-Point Step Size Gradient Methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [\[CrossRef\]](#)
- Li, Y.; Liu, Z.; Hongwei, L. A subspace minimization conjugate gradient method based on conic model for unconstrained optimization. *Comput. Appl. Math.* **2019**, *38*, 16. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Zhao, T.; Liu, H.; Liu, Z. New subspace minimization conjugate gradient methods based on regularization model for unconstrained optimization. *Numer. Algorithms* **2020**, *87*, 1501–1534. [\[CrossRef\]](#)
- Andrei, N. An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. *Numer. Algorithms* **2014**, *65*, 859–874. [\[CrossRef\]](#)
- Yang, Y.; Chen, Y.; Lu, Y. A subspace conjugate gradient algorithm for large-scale unconstrained optimization. *Numer. Algorithms* **2017**, *76*, 813–828. [\[CrossRef\]](#)
- Li, M.; Liu, H.; Liu, Z. A new subspace minimization conjugate gradient method with non-monotone line search for unconstrained optimization. *Numer. Algorithms* **2018**, *79*, 195–219. [\[CrossRef\]](#)
- Yao, S.; Wu, Y.; Yang, J.; Xu, J. A Three-Term Gradient Descent Method with Subspace Techniques. *Math. Probl. Eng.* **2021**, *2021*, 8867309.
- Yuan, Y. A review on subspace methods for nonlinear optimization. *Proc. Int. Congr. Math.* **2014**, 807–827.
- Wei, Z.; Yao, S.; Liu, L. The convergence properties of some new conjugate gradient methods. *Appl. Math. Comput.* **2006**, *183*, 1341–1350. [\[CrossRef\]](#)
- Yuan, Y. A modified BFGS algorithm for unconstrained optimization. *IMA J. Numer. Anal.* **1991**, *11*, 325–332. [\[CrossRef\]](#)
- Dai, Y.; Yuan, J.; Yuan, Y.X. Modified two-point stepsize gradient methods for unconstrained optimization. *Comput. Optim. Appl.* **2002**, *22*, 103–109. [\[CrossRef\]](#)
- Dai, Y.H.; Kou, C.X. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *Soc. Ind. Appl. Math. J. Optim.* **2013**, *23*, 296–320. [\[CrossRef\]](#)
- Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [\[CrossRef\]](#)

- 
25. Andrei, N. An unconstrained optimization test functions collection. *Adv. Model. Optim.* **2008**, *10*, 147–161.
  26. 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](#)]