

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

A Regularized Limited Memory Subspace Minimization Conjugate Gradient Method for Unconstrained Optimization

Wumei Sun Xidian University
Hongwei Liu (▲ hwliuxidian@163.com) Xidian University
Zexian Liu Guizhou University

Research Article

Keywords: Limited memory, Subspace minimization conjugate gradient method, Orthogonality, Regularization model, Quasi-Newton method

Posted Date: January 5th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2427275/v1

License: (c) This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Numerical Algorithms on June 19th, 2023. See the published version at https://doi.org/10.1007/s11075-023-01559-0.

A Regularized Limited Memory Subspace Minimization Conjugate Gradient Method for Unconstrained Optimization

Wumei Sun¹ · Hongwei Liu¹ · Zexian Liu²

Received: date / Accepted: date

Abstract In this paper, based on the limited memory techniques and subspace minimization conjugate gradient (SMCG) methods, a regularized limited memory subspace minimization conjugate gradient method is proposed, which contains two types of iterations. In SMCG iteration, we obtain the search direction by minimizing the approximate quadratic model or approximate regularization model. In RQN iteration, combined with regularization technique and BFGS method, a modified regularized quasi-Newton method is used in the subspace to improve the orthogonality. Moreover, some simple acceleration criteria and an improved tactic for selecting the initial stepsize to enhance the efficiency of the algorithm are designed. Additionally, an generalized nonmonotone line search is utilized and the global convergence of our proposed algorithm is established under mild conditions. Finally, numerical results show that, the proposed algorithm has a significant improvement over ASMCG_PR and is superior to the particularly well-known limited memory conjugate gradient software packages CG_DESCENT (6.8) and CGOPT(2.0) for the CUTEr library.

Keywords Limited memory \cdot Subspace minimization conjugate gradient method \cdot Orthogonality \cdot Regularization model \cdot Quasi-Newton method

Mathematics Subject Classification (2000) 49M37 · 65K05 · 90C30

1 Introduction

Consider problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable nonlinear function.

Wumei Sun E-mail: sunwumei1992@126.com Hongwei Liu ⊠ E-mail: hwliuxidian@163.com Zexian Liu

E-mail: liuzexian2008@163.com

¹ School of Mathematics and Statistics, Xidian University, Xi'an 710126, China

² School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China

Throughout the article, we use the following notations. $s_{k-1} = x_k - x_{k-1}$, $f_k = f(x_k)$, $g_k = g(x_k)$, $y_{k-1} = g_k - g_{k-1}$, $\|\cdot\|$ represents the Euclidean norm and λ_{\max} denotes the maximum eigenvalue. Moreover, $dist\{x, S\} = inf\{\|y - x\|, y \in S\}$, where $x \in \mathbb{R}^n$ and $S \in \mathbb{R}^n$.

Nonlinear conjugate gradient (CG) method is a well-known method for solving the problem (1.1), which main iteration is

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \cdots,$$
(1.2)

where x_k is the kth iteration point, $\alpha_k > 0$ is the stepsize and d_k is the search direction obtained by

$$d_0 = -g_0, \ d_k = -g_k + \beta_k d_{k-1}, \ k \ge 1,$$
(1.3)

where g_k is the gradient of $f(x_k)$ and β_k is the conjugate parameter.

It is shown in theory that the convergence and numerical performance variation of different CG methods depend on the selection of conjugate parameters. Some very classical choices of the conjugate parameter β_k are Fletcher-Reeves(FR) [9], Polak-Ribière-Polyak(PRP) [30,31], Dai-Yuan(DY) [7] and Hestenes-Stiefel(HS) [16], and are given by

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

CG algorithms have evolved considerably, and some well-known CG packages such as CG_DESCENT [12, 14] and CGOPT [5] have been proposed in recent years. Other recent related studies on nonlinear CG algorithms can be found in [4,13].

The subspace minimization conjugate gradient (SMCG) algorithm, as a generalization of the CG algorithm, has received much attention from scholars [1, 37], which can be traced back to the work of Yuan and Stoer [39]. The search direction of SMCG method is obtained by minimizing the following problem:

$$\min_{d \in \Omega_k} \quad g_k^T d + \frac{1}{2} d^T B_k d, \tag{1.4}$$

where Ω_k is a subspace spanned by the vectors g_k and s_{k-1} , i.e., $\Omega_k = Span\{g_k, s_{k-1}\}$, and $B_k \in \mathbb{R}^{n \times n}$ is an approximation of Hessian matrix, which is positive definite and symmetric. Then the search direction dis given by

$$d = ug_k + vs_{k-1},\tag{1.5}$$

where u and v are both real parameters. Substituting (1.5) to (1.4) and combined with the standard secant equation $B_k s_{k-1} = y_{k-1}$, formula (1.4) is reorganized as follows:

$$\min_{u,v \in \mathbb{R}} \begin{pmatrix} \|g_k\|^2 \\ g_k^T s_{k-1} \end{pmatrix}^T \begin{pmatrix} u \\ v \end{pmatrix} + \frac{1}{2} \begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} \rho_k & g_k^T y_{k-1} \\ g_k^T y_{k-1} & s_{k-1} y_{k-1} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$
(1.6)

where $\rho_k \approx g_k^T B_k g_k$.

On the basis of the Barzilai-Borwein(BB) method [2], Dai and Kou [6] proposed an effective BBCG3 method for strictly convex quadratic minimization problem. Afterwards, based on BBCG3 method, Liu and Liu [26] proposed SMCG_BB method for solving general unconstrained optimization problems.

Liu and Liu [26] proposed SMCG_BB method, which extended BBCG3 method to solve general unconstrained optimization problems. Motivated by SMCG_BB method, some efficient SMCG methods [20,21, 36,42] were later proposed, among which the method based on the regularization model presented by Zhao et al. [42] is the best in the numerical performance.

The nonlinear CG method is very effective for unconstrained optimization problems. However, the convergence of the algorithm can be very slow for some ill-posed problems and even for quadratic problems with very small dimensions, which may be due to the loss of orthogonality [15]. Hager and Zhang [15] pointed out theoretically that the generated successive gradients either in the CG method or the L-BFGS method for the quadratic test problem should be orthogonal. Yet, Hager and Zhang [15] observed that, when solving the quadratic strictly convex minimization problem PALMER1C in the CUTEr library [10], the CG method loses orthogonality due to the rounding errors, while L-BFGS method preserves the orthogonality. In view of this, they developed the limited memory CG method (CG_DESCENT(6.8)) to correct the possible loss of orthogonality in ill conditioned optimization problems. For the test problems in the CUTEr library [10], their performance results indicated that CG_DESCENT(6.8) has an significant improvement over their previously proposed package CG_DESCENT(5.3).

Although CG_DESCENT(6.8) [15] is an efficient method for unconstrained optimization, it still suffers from the following shortcomings:

(i) In the numerical implementation, the AWolfe line search [14] utilized in the algorithm CG_DESCENT(6.8) does not guarantee global convergence.

(ii) CG_DESCENT(6.8) contains the following three pre-conditioners, corresponding to three different iterations:

$$P_k = I, \ P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T, \ P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T + \sigma_k \bar{Z}_k \bar{Z}_k^T,$$
(1.7)

where σ_k is determined by (4.2) of [15], \hat{B}_{k+1} , Z_k and \bar{Z}_k are given by the matrices in literature [15]. These three pre-conditioners make the algorithm CG_DESCENT(6.8) look complex.

(iii) In the convergence analysis, the algorithm CG_DESCENT(6.8) needs to impose the following assumptions on the pre-conditioners:

$$\|P_k\| \le \gamma_0, \ g_{k+1}^T P_k g_{k+1} \ge \gamma_1 \|g_{k+1}\|^2, \ d_k^T P_k^{-1} d_k \ge \gamma_2 \|d_k\|^2,$$
(1.8)

where $\gamma_0 > 0$, $\gamma_1 > 0$ and $\gamma_2 > 0$. These assumptions are comparatively strict and difficult to be verified in actual practice.

To address the above-mentioned shortcomings, Liu et al. [27] presented an improved Dai – Kou CG algorithm called CGOPT(2.0), which combines limited memory technology and Dai-Kou CG method. In CGOPT(2.0) [27], they utilized a modified quasi-Newton method to restore the lost orthogonality, and established the convergence of CGOPT(2.0) with fewer assumptions. Some numerical experiments indicated that CGOPT(2.0) is better than the famous CG software package CG_DESCENT(6.8) [15].

In view of the above discussion, a regularized limited memory subspace minimization conjugate gradient method on the basis of SMCG method and limited memory technique is studied in this paper. To recover orthogonality, we propose a modified regularized quasi-Newton method. The major contributions of this paper are the following.

- 1. A regularized limited memory subspace minimization conjugate gradient algorithm is proposed, which combines limited memory technology and SMCG method.
- 2. Based on the idea of regularization and BFGS method, an improved regularized quasi-Newton method is exploited to improve orthogonality.
- 3. Some simple acceleration criteria and an improved initial stepsize selection strategy are designed to enhance the efficiency of the algorithm. Additionally, an generalized nonmonotone line search condition is presented, which may be regarded as an extension of the Zhang-Hager's [41] nonmonotone line search.
- 4. The convergence of the method is built under mild conditions and the corresponding numerical performance shows that the new method is much more effective than the existing methods.

The structure of the paper is as follows. In Section 2, we describe the detail of the regularized limited memory subspace minimization conjugate gradient algorithm, including the direction selection of SMCG iteration and regularized Quasi-Newton iteration and an effective acceleration technique. Moreover, the decision of the initial step size and the generalized nonmonotone Wolfe line search are also given in this section. In Section 3, some important properties of the search direction are analyzed and the global convergence of the proposed algorithm is established. Conclusions are given in the last section.

2 A Regularized Limited Memory Subspace Minimization Conjugate Gradient Algorithm

In the section, combining the idea of subspace minimization and regularization quasi-Newton method, we present a regularized limited memory subspace minimization conjugate gradient algorithm. Firstly, we give the choices of search direction under different iterations. Subsequently, we develop a very effective acceleration technique, a modified initial step selection strategy and generalized nonmonotonic line search technology to optimize the performance of the proposed algorithm. Finally, the details of algorithm RL_SMCG are described.

2.1 Direction Selection of SMCG Iteration and Regularized Quasi-Newton Iteration

The regularized limited memory subspace minimization conjugate gradient method mainly contains two kinds of iterations which are SMCG iteration and regularized quasi-Newton(RQN) iteration, respectively. Furthermore, the search direction derivation of the two iterations is also different.

2.1.1 SMCG iteration

The search direction selection of SMCG iteration is closely related to the properties of the objective function f(x) at the iteration point x_k . By reference [3,38], defined

$$t_{k} = \left| 2 \left(f_{k-1} - f_{k} + g_{k}^{T} s_{k-1} \right) / \left(s_{k-1}^{T} y_{k-1} \right) - 1 \right|,$$
(2.1)

to describe how f(x) approaches a quadratic function on a line segment between x_{k-1} and x_k . Literature [24] indicates that if the condition

$$t_k \le \xi_4 \text{ or } (t_k \le \xi_5 \text{ and } t_{k-1} \le \xi_5),$$
 (2.2)

is satisfied, where $\bar{\xi}_4$ and $\bar{\xi}_5$ are the smaller positive constants and $\bar{\xi}_4 < \bar{\xi}_5$, f(x) may be near to a quadratic function on a line between x_{k-1} and x_k . Moreover, According to [32], we know that if the following condition

$$\bar{\xi}_1 \le \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} \le \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \le \bar{\xi}_2,$$
(2.3)

is satisfied, then the condition number of the Hessian matrix of the normal function may be not very large, here $\bar{\xi}_1$ and $\bar{\xi}_2$ are positive constants.

Similar to [42], based on some certain properties of the function f(x) at the current point x_k , we derive different search direction by dividing it into the following four cases.

(i) If the condition (2.3) is satisfied while the condition (2.2) are not, this implies that the quadratic model may not be able to approach the objective function f(x) well at the present iteration point x_k . Then, search direction d_k will be obtained by minimizing the following cubic regular subproblem, i.e.

$$\min_{d_k \in \Omega_k} m_k \left(d_k \right) = d_k^T g_k + \frac{1}{2} d_k^T B_k d_k + \frac{1}{3} \sigma_k \left\| d_k \right\|_{B_k}^3,$$
(2.4)

where Ω_k is a subspace spanned by the vectors g_k and s_{k-1} , $B_k \in \mathbb{R}^{n \times n}$ is an approximation of Hessian matrix, which is positive definite and symmetric and satisfying the secant condition $B_k s_{k-1} = y_{k-1}$, $\sigma_k \ge 0$ is an adaptive regularization parameter obtained from interpolation condition and d_k is determined by

$$d_k = u_k g_k + v_k s_{k-1}, (2.5)$$

where v_k and u_k are parameters to be established. Obviously, we could obtain (2.4) by giving (1.4) a weighted regularization term $\frac{1}{3}\sigma_k \|d_k\|_{B_k}^3$. Substituting (2.5) to (2.4), it is easy to obtain that (2.4) is equivalent to

$$\min_{u_k, v_k \in R} \begin{pmatrix} \|g_k\|^2 \\ g_k^T s_{k-1} \end{pmatrix}^T \begin{pmatrix} u_k \\ v_k \end{pmatrix} + \frac{1}{2} \begin{pmatrix} u_k \\ v_k \end{pmatrix}^T \bar{B}_k \begin{pmatrix} u_k \\ v_k \end{pmatrix} + \frac{\sigma_k}{3} \left\| \begin{pmatrix} u_k \\ v_k \end{pmatrix} \right\|_{\bar{B}_k}^3.$$
(2.6)

where $\bar{B}_k = \begin{pmatrix} \rho_k & g_k^T y_{k-1} \\ g_k^T y_{k-1} & s_{k-1} y_{k-1} \end{pmatrix}$ is a positive definite and symmetric matrix, ρ_k is an estimate of $g_k^T B_k g_k$. Similar to BBCG3 [6], we also use $\frac{3}{2} \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} I$ to estimate B_k in the term ρ_k , which means $\rho_k = \frac{3}{2} \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \|g_k\|^2$. Then, by solving problem (2.6) we obtain the following solutions about u_k and v_k :

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} \frac{1}{(1+\sigma_k(\varpi^*))\Delta_k} \left(g_k^T y_{k-1} g_k^T s_{k-1} - s_{k-1}^T y_{k-1} \|g_k\|^2 \right) \\ \frac{1}{(1+\sigma_k(\varpi^*))\Delta_k} \left(g_k^T y_{k-1} \|g_k\|^2 - \rho_k g_k^T s_{k-1} \right) \end{pmatrix},$$
(2.7)

among them,

$$\Delta_k = \begin{vmatrix} \rho_k & g_k^T y_{k-1} \\ g_k^T y_{k-1} & s_{k-1} y_{k-1} \end{vmatrix} = \rho_k s_{k-1} y_{k-1} - (g_k^T y_{k-1})^2 > 0,$$
(2.8)

 σ_k and π^* are the same as those in literature [42], which will not be repeated here.

(ii) If both conditions (2.3) and (2.2) hold, this indicates that the objective function f(x) may approach the quadratic model at the current iteration point x_k . Since that is the case, let $\sigma_k = 0$, i.e. we consider deriving the search direction by solving the minimization problem (1.6). Like (i), we choose $\rho_k = \frac{3}{2} \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \|g_k\|^2$

$$\begin{pmatrix} \bar{u}_k \\ \bar{v}_k \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta_k} (g_k^T y_{k-1} g_k^T s_{k-1} - s_{k-1}^T y_{k-1} \|g_k\|^2) \\ \frac{1}{\Delta_k} (g_k^T y_{k-1} \|g_k\|^2 - \rho_k g_k^T s_{k-1}) \end{pmatrix},$$
(2.9)

here the search direction is calculated by $d_k = \bar{u}_k g_k + \bar{v}_k s_{k-1}$, where \bar{u}_k and \bar{v}_k are determined by (2.9).

(iii) If condition (2.3) is not satisfied and the conditions

$$\left|g_{k}^{T}y_{k-1}g_{k}^{T}s_{k-1}\right| \leq \bar{\xi}_{3}s_{k-1}^{T}y_{k-1}\|g_{k}\|^{2} \text{ and } s_{k-1}^{T}y_{k-1} \geq \bar{\xi}_{1}\|s_{k-1}\|^{2},$$
(2.10)

are satisfied, where $0 \leq \bar{\xi}_3 \leq 1$, the condition number of the Hessian matrix may be lager, hence the search direction obtained in cases (i) and (ii) may not be better. However, the condition (2.10) can ensure sufficient descent and linear growth in HS conjugate gradient method. Moreover, because of the finite termination nature of the HS conjugate gradient method for solving exact convex quadratic minimization problems, this choice of direction allows for faster convergence of the algorithm. Then, in this case, the search direction is determined by (1.3) and $\beta_k = \beta_k^{HS}$.

(iv) If neither condition (2.3) nor (2.10) holds, then we pick the following direction, i.e.:

$$d_k = -g_k. \tag{2.11}$$

In summary, the search direction in the SMCG iteration can be described as in the following:

$$d_{k} = \begin{cases} u_{k}g_{k} + v_{k}s_{k-1}, & \text{if (2.3) holds and (2.2) does not hold,} \\ \bar{u}_{k}g_{k} + \bar{v}_{k}s_{k-1}, & \text{if (2.3) holds and (2.2) holds,} \\ -g_{k} + \beta_{k}^{HS}d_{k-1}, & \text{if (2.3) does not hold and (2.10) holds,} \\ -g_{k}, & \text{if neither (2.3) nor (2.10) holds,} \end{cases}$$

$$(2.12)$$

where u_k and v_k are determined by (2.7); \bar{u}_k and \bar{v}_k are determined by (2.9).

If the successive gradients have orthogonality or the lost orthogonality is restored, the algorithm performs SMCG iteration. On the contrary, if the orthogonality is lost, the iteration will turn to the following regularized quasi-Newton iteration to improve the orthogonality.

2.1.2 Regularized Quasi-Newton(RQN) iteration

When the successive gradients lose their orthogonality, the iteration switches from SMCG iteration to RQN iteration. In other words, a modified regularized BFGS algorithm in subspace S_k is proposed to restore the orthogonality, where S_k is a subspace generated by the following limited memory m search directions

$$\mathcal{S}_k = span \left\{ d_{k-1}, d_{k-2}, \cdots, d_{k-m} \right\},\,$$

where m > 0 and m is the number of limited memory. In this article, the limited memory m selected in our algorithm does not exceed 11. Then, as soon as orthogonality is corrected, the RQN iteration is terminated and the SMCG iteration is triggered immediately.

First, we introduce some preparations for turning to RQN iteration. Let $S_k \in \mathbb{R}^{n \times m}$ be a matrix which has columns consisting of $d_{k-1}, d_{k-2}, \dots, d_{k-m}$. In similar fashion to limited memory CG method [15], we also assume that columns of S_k are line-independent. Let the QR factorization of S_k be $S_k = Z_k \bar{R}_k$, where the columns of $Z_k \in \mathbb{R}^{n \times m}$ form the normal orthogonal bases for subspace S_k and $\bar{R}_k \in \mathbb{R}^{m \times m}$ is the upper triangular matrix with positive diagonal terms.

If g_k is included almost in subspace S_k , then we think that the orthogonality property of the algorithm may be lost. In this case, we interrupt the SMCG iteration and move to minimize the objective function in the subspace S_k :

$$\min_{z \in \mathcal{S}} f(x_k + z). \tag{2.13}$$

The solution to the subspace problem (2.13) will improve the orthogonality and guide us to a suitable search direction that will lead us out of the subspace S_k . Similar to [15], we utilize the distance from g_k to subspace S_k to judge whether orthogonality is lost. If the condition

$$\operatorname{dist}\left\{g_k, \mathcal{S}_k\right\} \le \tilde{\eta}_0 \|g_k\| \tag{2.14}$$

is satisfied, where $0 < \tilde{\eta}_0 < 1$ and $\tilde{\eta}_0$ is small, we think g_k is almost contained in S_k , it means that the orthogonality of the successive gradients has lost. Then, we switch to RQN iteration to solve the subspace problem (2.13) until the gradient is nearly orthogonal enough to the subspace to meet the condition

$$\operatorname{dist}\left\{g_k, \mathcal{S}_k\right\} \ge \tilde{\eta}_1 \|g_k\|,\tag{2.15}$$

where $0 < \tilde{\eta}_0 < \tilde{\eta}_1 < 1$. At this time, the algorithm iteration will go away subspace S_k and turn to the SMCG iteration. Because the column of Z_k is the orthonormal basis of S_k , it's not hard to know from the definition of dist $\{g_k, S_k\}$ that (2.14) and (2.15) can be expressed as

$$(1 - \tilde{\eta}_0^2) \|g_k\|^2 \le \|Z_k^T g_k\|^2,$$
 (2.16)

and

$$\left(1 - \tilde{\eta}_{1}^{2}\right) \left\|g_{k}\right\|^{2} \ge \left\|Z_{k}^{T} g_{k}\right\|^{2}.$$
 (2.17)

In [15], Hager and Zhang utilized the limited memory BFGS (L-BFGS) [22, 28] method to solve the subspace problem (2.13) for restoring the orthogonality, and achieved better numerical results. However, it should be noted that the convergence analysis of the limited memory CG method [15] requires imposing strict assumptions (1.8) on the preprocessors (1.7). Because the dimension m of the chosen subspace S_k is usually small and when orthogonality is lost, the properties of the function at the iteration point maybe not very good. Based on these, we consider a regularized L-BFGS method in the subspace S_k for solving the subproblem (2.13).

The search direction of general quasi-Newton method [40] for unconstrained optimization (1.1) is the form of $d_k = -B_k^{-1}g_k$, where B_k is a positive definite and symmetric approximation to the Hessian matrix. As one of the most popular methods of quasi-Newton method, L-BFGS method stores the approximate Hessian matrix of the objective function using small memory and computes the search direction d_k using the nearest m vector pairs of $(s_{k-i}, y_{k-i}), i = 0, 1, ..., m - 1$.

$$\left(\nabla^2 f(x_k) + \mu I\right) d_k = -\nabla f(x_k), \qquad (2.18)$$

where $\mu > 0$ is referred to as the regularized parameter. The regularized Newton method [35] generally defaults to a step size of 1, and global convergence is guaranteed by controlling the parameter μ_k . However, as a type of Newton method, the regularized Newton method in [35] must solve the Hessian matrix of f which is particularly computationally complex. To address this drawback, some scholars proposed the regularized limited memory BFGS-type method [33,23] for solving unconstrained optimization problems, i.e. the search direction d_k is the solution of the following equations

$$(B_k + \mu I) d_k = -\nabla f(x_k), \qquad (2.19)$$

where matrix B_k is an approximate Hessian determined by a particular quasi-Newton method. Regularization technology can effectively improve the efficiency of quasi-Newton method in solving ill-conditioned problems. Nevertheless, when computing B_k by the L-BFGS method, it is very hard to calculate $(B_k + \mu I)^{-1}$. Hence, motivated by [34], we present a regularized quasi-Newton method which combines the BFGS method with the regularized technique to improve orthogonality in the m-dimensional subspace S_k . In this paper, we consider $B_k + \mu I$ as an approximation of $\nabla^2 f(x_k) + \mu I$. Because the matrix B_k is the approximate Hessian of $f(x_k)$ and $B_k + \mu I$ can be used as an approximate Hessian of $f(x_k) + \frac{\mu}{2} ||x||^2$. At this point, we utilize $(s_k, y_k(\mu))$ instead of (s_k, y_k) , where

$$y_k(\mu) = (\nabla f(x_{k+1}) + \mu x_{k+1}) - (\nabla f(x_k) + \mu x_k) = y_k + \mu s_k$$

Note that the regularized BFGS method stores as many vector pairs as the traditional BFGS method and hence it does not require additional memory.

In [19], a effective BFGS quasi-Newton method for solving nonconvex unconstrained minimization was proposed by Li and Fukushima [19], in which the matrix B_{k+1} is updated by

$$B_{k+1} = \begin{cases} B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, & \text{if } \frac{s_k^T y_k}{\|s_k\|^2} > v \|g_k\|^{\alpha}, \\ B_k, & \text{otherwise }, \end{cases}$$

where v > 0 and $\alpha > 0$. Some recent advances about modified BFGS method can be found in [18,11,34].

Inspired by the quasi-Newton methods described above, we propose an improved regularized BFGS method to solve the subproblem (2.13) in subspace S_k .

Remark 1. In what follows, the variables with hats belong to subspace S_k , distinguished from the ones found in the full space \mathbb{R}^n .

Let $\hat{x} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_m)^T \in \mathbb{R}^m$. The subproblem (2.13) can be expressed as

$$\min_{\hat{x}\in\mathbb{R}^m} f(x_k + \hat{x}_1 d_{k-1} + \hat{x}_2 d_{k-2} + \dots + \hat{x}_m d_{k-m}).$$
(2.20)

Similar to [27], because the regularized quasi-Newton directions in the subspace S_k always transform to the full space \mathbb{R}^n and QR decomposition of matrix S_k , we can obtain $d_k = Z_k \hat{d}_k$, $\hat{g}_k = Z_k^T g_k$, $\hat{y}_k = Z_k^T y_k$, $\hat{s}_k^T \hat{y}_k = s_k^T y_k$, $\|\hat{s}_k\|^2 = \|s_k\|^2$ and $\hat{f}_k = f_k$.

Let $B_k(\mu) = B_k + \mu I$, then inspired by Li and Fukushima [19], we develop an improved regularized BFGS method to solve the above subproblem (2.20) with a search direction of the form

$$\hat{d}_{k+1} = -\hat{B}_{k+1}^{-1}(\mu)\hat{g}_{k+1}, \qquad (2.21)$$

where $\hat{B}_{k+1}(\mu)$ is given by

$$\hat{B}_{k+1}(\mu) = \begin{cases} \hat{B}_k(\mu) - \frac{\hat{B}_k(\mu)\hat{s}_k\hat{s}_k^T\hat{B}_k(\mu)}{\hat{s}_k^T\hat{B}_k(\mu)\hat{s}_k} + \frac{\hat{y}_k(\mu)\hat{y}_k^T(\mu)}{\hat{s}_k^T\hat{y}_k(\mu)}, & \text{if } \text{mod}(k,l) \neq 0 \text{ and } \frac{\hat{s}_k^T\hat{y}_k(\mu)}{\hat{s}_k^T\hat{s}_k} \geq \upsilon \text{ [27]}, \\ \hat{I}, & \text{otherwise }, \end{cases}$$

$$(2.22)$$

where v > 0, $\operatorname{mod}(k, l) \neq 0$ represents the remainder for k modulo n, $\hat{y}_k(\mu) = \hat{y}_k + \mu \hat{s}_k$ and $\mu > 0$ is an important regularized parameter. The condition $\operatorname{mod}(k, l) \neq 0$ means the matrix $\hat{B}_k(\mu)$ will be reset to the identity matrix \hat{I} after updating l times, which ensures the good convergence of the algorithm. In the paper, we set $l = \max(m^2, 20)$. Obviously, $\hat{s}_k^T \hat{y}_k(\mu) > 0$, and as soon as the matrix $\hat{B}_k(\mu)$ is symmetric and positive definitive, it is not hard to prove that the matrix $\hat{B}_{k+1}(\mu)$ is symmetric and positive definitive.

As a very important regularization parameter, μ is closely related to the convergence analysis of the regularized BFGS method. In this paper, the idea of the trust-region radius is used to find the suitable search direction by controlling μ , in other words, The ratio of objective function value reduction to model function value reduction is utilized. Then, give the definition of a ratio function $r_k(\hat{d}_k, \mu)$ as follows

$$r_k(\hat{d}_k,\mu) = \frac{\hat{f}(x_k) - \hat{f}(x_k + \alpha_k \hat{d}_k)}{\hat{f}(x_k) - \hat{q}_k(\hat{d}_k,\mu)},$$
(2.23)

where $\hat{q}_k : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is a function of the form

$$\hat{q}_k(\hat{d}_k,\mu) = \hat{f}(x_k) + \alpha_k \hat{g}_k^T \hat{d}_k + \frac{1}{2} \alpha_k^2 \hat{d}_k^T \hat{B}_k(\mu) \hat{d}_k.$$
(2.24)

Then, if the ratio function $r_k(\hat{d}_k, \mu)$ is relatively large, this means that compared with the reduction of the model function, the reduction of the objective function is large enough, we choose to reduce the parameter μ . On the flip side, if the ratio function $r_k(\hat{d}_k, \mu)$ is relatively small, i.e., $\hat{f}(x_k) - \hat{f}(x_k + \alpha_k \hat{d}_k)$ is small, we will increase μ . In addition, to ensure that the algorithms converge well, we limit μ to an interval, i.e. $0 < \mu_{\min} < \mu < \mu_{\max}$. In general, if the next iteration point is closer to the current iteration point, the reduction of the function value may not be obvious. At this time, we hope to get a new iteration point by modifying the search direction, then the search direction improved by regular parameter μ may be a good choice. Therefore, if $\|\hat{s}_k\|^2 \leq \hat{\tau}$ ($\hat{\tau} > 0$), our choice and update of μ are as follows:

$$\mu_{k+1} = \begin{cases} \max\left\{\mu_{\min}, \sigma_1 \mu_k\right\}, & \text{if } r_k(\hat{d}_k, \mu) \ge \sigma_3, \\ \min\left\{\mu_{\max}, \sigma_2 \mu_k\right\}, & \text{otherwise,} \end{cases}$$
(2.25)

where $0 < \sigma_1 \le 1$, $\sigma_2 > 1$ and $0 < \sigma_3 \le 1$. Otherwise, we choose $\mu = 0$, i.e., the regularized BFGS method is reduced to a general BFGS method.

Remark 2. In order to simplify the symbol and facilitate writing, we still record the updated symbol μ_{k+1} as μ .

In the process of algorithm implementation, the search direction (2.21) in subspace S_k always converts to the full space \mathbb{R}^n at each RQN iteration, i.e.,

$$d_{k+1} = -P_k g_{k+1}, (2.26)$$

where

$$P_k = Z_k \hat{B}_{k+1}^{-1}(\mu) Z_k^T \tag{2.27}$$

and $\hat{B}_{k+1}(\mu)$ is given by (2.22).

In Section 3, we will show that matrices $\hat{B}_{k+1}(\mu)$ and P_k have some good properties in the RQN iteration, which is critical for the convergence analysis.

2.2 An Effective Acceleration Technique

In order to optimize the performance of the algorithm, Sun et al. [32] proposed an acceleration technique, which replaces (1.2) with the following new iterative form

$$x_{k+1} = x_k + \bar{\eta}_k \alpha_k d_k, \tag{2.28}$$

where $\bar{\eta}_k \geq 0$ is an acceleration parameter obtained from an interpolation function. In view of the numerical effect of the acceleration technique, our algorithm also takes it into account. Similar to reference [32], we minimize the following interpolation function to get the acceleration parameter $\bar{\eta}_k$:

$$\bar{\eta}_k = \arg\min \ q(\varphi_k(\bar{\eta})), \tag{2.29}$$

where $\bar{\eta} \ge 0$, $\varphi_k(\bar{\eta}) = f(x_k + \bar{\eta}\alpha_k d_k)$, and $q(\varphi_k(\bar{\eta}))$ represents the interpolation function by. In the paper, we consider minimizing the quadratic interpolation function [29] $q(\varphi_k(0), \varphi'_k(0), \varphi'_k(1))$, then,

$$\bar{\eta}_k = \arg\min \ q(\varphi_k(0), \varphi'_k(0), \varphi'_k(1)),$$
(2.30)

By minimizing (2.30) we have

$$\bar{\eta}_k = -\frac{\bar{a}_k}{\bar{b}_k}, \ \bar{b}_k \ge \bar{\epsilon},\tag{2.31}$$

where $\bar{a}_k = \alpha_k g_k^T d_k$, $\bar{b}_k = \alpha_k (g_{\bar{z}} - g_k)^T d_k$, $g_{\bar{z}} = \nabla f(\bar{z})$, $\bar{z} = x_k + \alpha_k d_k$ and $\bar{\epsilon} > 0$ is a small constant.

We propose the following acceleration criterion, which is simpler than the rule in reference [32], that is

$$\bar{b}_k \ge \bar{\epsilon}, \ \|s_{\bar{z}}\|^2 \le \bar{\tau}, \ \|g_k\|^2 \le \hat{\tau}, \ |\bar{t}_{k+1}| < \bar{c}, \ \text{and} \ |s_k^T g_{\bar{z}}| \ge \operatorname{Max}(\varsigma, \bar{\varsigma} \cdot \bar{b}_k)$$
(2.32)

where $\bar{\epsilon}, \bar{\tau}, \hat{\tau}, \bar{c}, \varsigma$ and $\bar{\varsigma}$ are all small positive constants, $\bar{b}_k = \alpha_k (g_{\bar{z}} - g_k)^T d_k$, $s_{\bar{z}} = \bar{z} - x_k$, $\bar{z} = x_k + \alpha_k d_k$, $|\bar{t}_{k+1}| = |\frac{2(f_k - f_{\bar{z}} + g_{\bar{z}}^T s_{\bar{z}})}{s_{\bar{z}}^T g_{\bar{z}}} - 1|$, $f_{\bar{z}} = f(\bar{z})$ and $g_{\bar{z}} = \nabla f(\bar{z})$. When the condition (2.32) holds, we accelerate the algorithm and update the relevant variables. In addition, one of the necessary conditions for successful acceleration is that the trial iteration point must satisfy the line search condition. Therefore, if the

algorithm accelerates successfully, update the iteration point x_{k+1} by using (2.28). Otherwise the algorithm acceleration fails and returns to the original algorithm, at which point $\bar{\eta}_k = 1$, update the iteration point x_{k+1} with (1.2).

In reference [32], the acceleration criterion is divided into three cases, which seems to be more complex, while our acceleration criterion has only one case and the form is simpler.

2.3 Choices of the Initial Stepsize and the Generalized Nonmonotone Wolfe Line Search

It is well known that the design of the search direction and the conditions of the line search are two critical factors which affect the efficiency of the line search algorithm. In this subsection, we will develop an improved nonmonotone Wolfe line search which can be regarded as an extension of the Zhang-Hager's [41] nonmonotone line search. In addition, an improved initial step selection strategy is designed.

For the sake of convenience, we express the one-dimensional line search function as

$$\phi_k(\alpha) = f(x_k + \alpha d_k), \alpha \ge 0.$$

The choice of the initial stepsize α_k^0 is of great importance for a line search in an optimization method. For the Newton-like methods, choosing the initial step $\alpha_k^0 = 1$ is important to speed up convergence. For the conjugate gradient methods, it is essential to use information from the current iteration of the problem to make initial guesses [29]. In the conjugate gradient method, there have been various ways to choose the initial stepsize, for example, see [5,12,15,29]. However, it did not have an agreement on which is the best. In particular, Hager and Zhang [15] select the initial step in CG_DESCENT as below:

$$\alpha_{k}^{0} = \begin{cases} \arg\min\bar{q} \left(\phi_{k} \left(0 \right), \phi_{k}' \left(0 \right), \phi_{k} \left(\bar{\tau}_{1} \alpha_{k-1} \right) \right), & \text{if } \phi_{k} \left(\bar{\tau}_{1} \alpha_{k-1} \right) \leq \phi_{k} \left(0 \right), \\ \bar{\tau}_{2} \alpha_{k-1}, & \text{otherwise,} \end{cases}$$
(2.33)

where $\bar{q}(\phi_k(0), \phi'_k(0), \phi_k(\tau_1\alpha_{k-1}))$ represents the interpolation function given by the three values $\phi_k(0)$, $\phi'_k(0)$ and $\phi_k(\tau_1\alpha_{k-1})$, $\bar{\tau}_1$ and $\bar{\tau}_2$ are positive parameters. In CGOPT, Dai and Kou [5] determined the initial stepsize in the following way:

$$\alpha_{k}^{0} = \begin{cases} \alpha & \text{if } \left|\phi_{k}\left(\alpha\right) - \phi_{k}\left(0\right)\right| / \left(\tau_{3} + \phi_{k}\left(0\right)\right) > \tau_{4}, \\ \arg\min\bar{q}\left(\phi_{k}\left(0\right), \phi_{k}'\left(0\right), \phi_{k}\left(\alpha\right)\right), \text{ otherwise,} \end{cases}$$
(2.34)

where $\alpha = \max \left\{ \tau_5 \alpha_{k-1}, -2 | f_k - f_{k-1} | / g_k^T d_k \right\}, \tau_3 > 0, \tau_4 > 0$ and $\tau_5 > 0$. Most recently, Liu and Liu [26] discussed the development a very effective initial stepsize selection strategy for SMCG method by combining the BB methods and the interpolation technique.

Based on the above research, we devise an improved strategy to obtain the initial stepsize. We first consider the initial stepsize for the search direction in the RQN iteration.

(i) Initial stepsize of the search direction (2.26) with $B_{k+1}(\mu) \neq I$.

Since the search direction \hat{d} is a quasi-Newton direction in the subspace S_k , then the initial stepsize $\alpha_k^0 = 1$ may be a good choice. Therefore, the trial initial stepsize can be stated as

$$\alpha_k^0 = \begin{cases} \hat{\alpha}_k, & \text{if } ((2.2) \text{ or } \varpi \le \tau_2) \text{ holds and } \bar{\alpha}_k > 0, \\ 1, & \text{otherwise,} \end{cases}$$
(2.35)

where

$$\hat{\alpha}_k = \min\{\max\{\bar{\alpha}_k, \alpha_{\min}\}, \alpha_{\max}\}, \ \bar{\alpha}_k = \min\bar{q}(\phi_k(0), \phi_k'(0), \phi_k(1)), \phi_k(1)\},$$

$$\varpi = |\phi_k(1) - \phi_k(0)| / (\tau_1 + \phi_k(0)), \ \tau_1 > 0, \ \tau_2 > 0 \text{ and } \alpha_{\max} > \alpha_{\min} > 0$$

Here, $\bar{q}(\phi_k(0), \phi'_k(0), \phi_k(1))$ is a quadratic interpolation function for $\phi_k(0), \phi'_k(0)$, and $\phi_k(1)$, and α_{\max} and α_{\min} represent two positive constants.

(ii) Initial stepsize of the search direction (2.26) with $B_{k+1}(\mu) = I$.

$$\alpha_k^0 = \begin{cases} \hat{\alpha}_k, & \text{if } ((2.2) \text{ or } \varpi \le \tau_2) \text{ holds and } \bar{\alpha}_k > 0, \\ \bar{\alpha}_k, & \text{otherwise,} \end{cases}$$
(2.36)

where

$$\bar{\bar{\alpha}}_{k} = \begin{cases} \max\{\min\{\alpha_{k}^{BB_{2}}, \alpha_{\max}\}, \alpha_{\min}\}, & \text{if } g_{k}^{T}s_{k-1} > 0, \\ \max\{\min\{\alpha_{k}^{BB_{1}}, \alpha_{\max}\}, \alpha_{\min}\}, & \text{if } g_{k}^{T}s_{k-1} \le 0, \end{cases}$$
(2.37)

For the initial stepsize of the search direction in the SMCG iteration. If the search direction d_k is calculated by (2.12) with $d_k \neq -g_k$, the initial stepsize is chosen in the same way as the RQN iteration, which is determined by (2.35). If the search direction d_k is given by (2.11), the initial stepsize is determined by

$$\alpha_k^0 = \begin{cases} \min\{\max\{\tilde{\tilde{\alpha}}_k, \alpha_{\min}\}, \alpha_{\max}\}, & \text{if (2.2) holds, } \|g_k\|^2 \le 1, \ d_{k-1} \ne -g_{k-1} \text{ and } \tilde{\tilde{\alpha}}_k > 0, \\ \bar{\tilde{\alpha}}_k, & \text{otherwise,} \end{cases}$$
(2.38)

where $\bar{\bar{\alpha}}_k$ is determined by (2.37) and $\tilde{\tilde{\alpha}}_k = \min q(\phi_k(0), \phi_k'(0), \phi_k(\bar{\bar{\alpha}}_k)).$

Next, we introduce a generalized line search condition, which can be regarded as a development of the Zhang-Hager's nonmonotone line search. We recall the nonmonotone line search introduced by Zhang and Hager [41]

$$f(x_k + \alpha_k d_k) \le C_k + \delta \alpha_k g_k^T d_k, \tag{2.39}$$

where

$$C_{k+1} = \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}}, \quad Q_{k+1} = \eta_k Q_k + 1, \tag{2.40}$$

 $0 < \delta < 1$, and $\eta_k \in [0,1]$. From (2.40), it is easy to see that C_{k+1} is a convex combination of f_{k+1} and C_k . If $C_0 = f(x_0)$, it is thus clear that C_k can be regard as a convex combination of the function values $f(x_0), f(x_1), \dots, f(x_k)$. It means that C_k can employ information about the known function values from the previous iteration. The Zhang-Hager's nonmonotone line search (2.39) is reduced to the standard Armijo line search condition when $\eta_k = 0$ for each k.

As it was reported in [41], the nonmonotone line search proposed by Zhang and Hager plays a crucial role in generating an appropriate stepsize compared to the monotone line search method. Based on (2.39) and (2.40), Huang et al. [17] presented a very effective nonmonotone line search technique, which can be regard as an extension of Zhang-Hager's nonmonotone line search, that is

$$C_{k+1} = \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \le C_k + \delta_k \alpha_k g_k^T d_k,$$
(2.41)

where $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\delta_{\max} < 1$, $0 < \delta_{\min} < (1 - \eta_{\max})\delta_{\max}$, $\delta_{\min} \le \delta_k \le \frac{\delta_{\max}}{Q_{k+1}}$ and Q_{k+1} is computed by (2.40).

Inspired by the previous discussion, we will study a generalized nonmonotone Wolfe line search technique based on (2.40) and (2.41). Considering the acceleration technique, the generalized nonmonotone Wolfe line search conditions are as follows:

$$C_{k+1} \le C_k + \delta_k \bar{\eta}_k \alpha_k g_k^T d_k, \qquad (2.42)$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k, \tag{2.43}$$

where $0 < \delta_{\min} < \delta_k < \delta_{\max} < 1$, $\sigma \in (0, 1)$, $Q_0 = 1$, $C_0 = f_0$, $\bar{\eta}_k$ is an acceleration parameter determined by (2.31), C_k and Q_k are updated as follows

$$C_{k+1} = \frac{\eta_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}, \ Q_{k+1} = \eta_k Q_k + 1, \ f(x_{k+1}) = f(x_k + \bar{\eta}_k \alpha_k d_k),$$
(2.44)

where $\eta_k \in [0, 1]$. Specially,

$$Q_1 = 2.0, \ C_1 = \min\{C_0, f_1 + 1.0\},$$
 (2.45)

when $k \ge 1$, C_{k+1} and Q_{k+1} are updated by (2.44), and η_k is given as

$$\eta_k = \begin{cases} 1, & \text{if } C_k - f_{k+1} > 0.95 |C_k| \text{ and } k > 100, \\ 0.9, & \text{otherwise.} \end{cases}$$
(2.46)

Here η_k is a parameter that controls the degree of non-monotonicity, referred to [25].

Furthermore, we demonstrate that the generalized nonmonotone Wolfe line search is an extension of the Zhang-Hager's nonmonotone Wolfe line search method. It follows from (2.42) that we get

$$f(x_k + \bar{\eta}_k \alpha_k d_k) \le (Q_{k+1} - \eta_k Q_k)C_k + Q_{k+1}\delta_k \bar{\eta}_k \alpha_k g_k^T d_k.$$

$$(2.47)$$

Since $Q_{k+1} - \eta_k Q_k = 1$, (2.42) is equivalent to

$$f(x_k + \bar{\eta}_k \alpha_k d_k) \le C_k + Q_{k+1} \delta_k \bar{\eta}_k \alpha_k g_k^T d_k, \qquad (2.48)$$

It is easy to see that if $\delta_k = \frac{\delta}{Q_{k+1}}$, nonmonotone line search condition (2.48) reduces to the Zhang-Hager's nonmonotone Wolfe line search condition (2.39). This means that the Zhang-Hager's nonmonotone Wolfe line search condition in [41] can be considered as a particular version of (2.42).

2.4 A Regularized Limited Memory Subspace Minimization Conjugate Gradient Algorithm(RL_SMCG)

In this subsection, we describe the regularized limited memory subspace minimization conjugate gradient algorithm in detail. As mentioned above, the regularized limited memory subspace minimization conjugate gradient algorithm is made of two kinds of iterations. The "state" in Algorithm 1 represents for the type of iteration, i.e., state= "SMCG" means that SMCG iteration will be carried out, and state= "RQN" means that RQN iteration will be performed.

Algorithm 1 RL_SMCG

Step 0. Chosen $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $\tilde{\eta}_0$, $\tilde{\eta}_1$, v, m, ξ_1 , ξ_2 , ξ_3 , ξ_4 , ξ_5 , σ_1 , σ_2 , σ_3 , μ_{\min} , μ_{\max} , τ , $\bar{\tau}$, \bar{c} , ς , $\bar{\varsigma}$, $\bar{\tau}_1$, τ_2 , δ_k , σ , IterRestart := 0, IterQuad := 0 and MinQuad. Set state = "SMCG" and k := 0. Step 1. If $||g_k||_{\infty} \le \varepsilon$, stop. Step 2. Compute the search direction. If (state = "SMCG"), then If k = 0, then $d_0 = -g_0$. elseif (IterQuad = MinQuad and IterQuad \neq IterRestart), set $d_k = -g_k$, IterQuad = 0, and IterRestart = 0. else Determine the search direction d_k by (2.12). end elseif (state = "RQN"), then Compute P_k by (2.27), and compute the search direction d_k by (2.26).

end

Step 3. Determine the corresponding initial step size α_k^0 from (2.35), (2.36) and (2.38) according to the different iteration directions in the Step 2.

Step 4. Determine a stepsize α_k satisfying the generalized nonmonotone Wolfe line search (2.42) and (2.43) with initial stepsize α_k^0 .

Step 5. Compute the trial iteration $\bar{z} = x_k + \alpha_k d_k$ and $g_{\bar{z}} = \nabla f(\bar{z})$. If $\|g_{\bar{z}}\|_{\infty} \leq \varepsilon$, then stop; otherwise, go to Step 6.

Step 6. Acceleration procedure.

If the condition (2.32) holds, then go to 6.1.

6.1. Compute $\bar{a}_k = \alpha_k g_k^T d_k$, $\bar{b}_k = \alpha_k (g_{\bar{z}} - g_k)^T d_k$ and $\bar{\eta}_k$ by (2.31).

6.2. Update the iteration point as $x_{k+1} = x_k + \bar{\eta}_k \alpha_k d_k$ and compute f_{k+1} and g_{k+1} .

6.3. If f_{k+1} satisfies (2.42) and g_{k+1} satisfies (2.43), go to Steps 8. Otherwise, go to Steps 7.

else

go to Steps 7.

end

Step 7. Update the variable as $x_{k+1} = x_k + \alpha_k d_k$. Compute f_{k+1} and g_{k+1} .

Step 8. Update restart conditions.

Step 9. Update Q_{k+1} and C_{k+1} with (2.44).

Step 10. Update iteration type.

```
If (state = "SMCG"), then

If (2.16) holds, then state = "RQN".

elseif (state = "RQN"), then

If (2.17) holds, then state = "SMCG".

end

Step 11. Set k := k + 1 and go to Step 1.
```

Remark 3. Notably, when the lost orthogonality is corrected, our algorithm terminates the RQN iteration and immediately calls the SMCG iteration. However, the limited memory CG method [15] first carries out the complex preprocessing CG iteration after the orthogonality is improved. This means that algorithm RL_SMCG is more simple compared to the limited memory CG method [15].

3 Convergence Analysis

In the section, we establish the global convergence of the algorithm RL_SMCG under the following assumptions and properties.

Define \mathcal{N} to be an open neighborhood of the level set $L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, where x_0 is an initial point.

Assumption 1 (i) The objective function f is continuously differentiable in \mathcal{N} and the level set is bounded from below. (ii) The gradient g of the objective function is Lipschitz continuous in \mathcal{N} , i.e., there exists a constant L > 0 such that $||g(x) - g(y)|| \le L ||x - y||, \forall x, y \in \mathcal{N}$.

Under these assumptions, we have the following several properties.

Lemma 1 Suppose that Assumption 1 holds. Then, for $\hat{B}_{k+1}(\mu)$ in (2.22), there exist three constants $\hat{\xi}_1 > 0$, $\hat{\xi}_2 > 0$ and $\hat{\xi}_3 > 0$ such that

$$\lambda_{\max}\left(\hat{B}_{k+1}(\mu)\right) \le \hat{\xi}_1, \ \lambda_{\max}\left(\hat{B}_{k+1}^{-1}(\mu)\right) \le \hat{\xi}_2, \ \left\|\hat{B}_{k+1}^{-1}(\mu)\right\| \le \hat{\xi}_3.$$

Proof We know that Z_k is a normal orthogonal basis of S_k and the dimension $m < +\infty$, hence we have $\xi_0 > 0$ such that $||Z_k|| \le \xi_0$. According to (2.22) and the property of the matrix norm in finite dimensional spaces, we can get that $\lambda_{\max}\left(\hat{B}_k(\mu)\right) = 1$ or

$$\lambda_{\max}\left(\hat{B}_{k+1}(\mu)\right) \leq \lambda_{\max}\left(\hat{B}_{k}(\mu)\right) + \lambda_{\max}\left(-\frac{\hat{B}_{k}(\mu)\hat{s}_{k}\hat{s}_{k}^{T}\hat{B}_{k}(\mu)}{\hat{s}_{k}^{T}\hat{B}_{k}(\mu)\hat{s}_{k}}\right) + \lambda_{\max}\left(\frac{\hat{y}_{k}(\mu)\hat{y}_{k}^{T}(\mu)}{\hat{s}_{k}^{T}\hat{y}_{k}(\mu)}\right) \qquad (3.1)$$
$$\leq \lambda_{\max}\left(\hat{B}_{k}(\mu)\right) + \frac{\hat{y}_{k}^{T}(\mu)\hat{y}_{k}(\mu)}{\hat{s}_{k}^{T}\hat{y}_{k}(\mu)}.$$

Further, by $\hat{y}_k(\mu) = \hat{y}_k + \mu \hat{s}_k, \ \mu > 0$, we get

$$\begin{aligned} \frac{\hat{y}_k^T(\mu)\hat{y}_k(\mu)}{\hat{s}_k^T\hat{y}_k(\mu)} &= \frac{\|\hat{y}_k\|^2 + \mu_k^2\|\hat{s}_k\|^2 + 2\mu\hat{s}_k^T\hat{y}_k}{\hat{s}_k^T\hat{y}_k + \mu\|\hat{s}_k\|^2} \\ &= \frac{\|\hat{y}_k\|^2 + \mu\hat{s}_k^T\hat{y}_k}{\hat{s}_k^T\hat{y}_k + \mu\|\hat{s}_k\|^2} + \frac{\mu\hat{s}_k^T\hat{y}_k + \mu_k^2\|\hat{s}_k\|^2}{\hat{s}_k^T\hat{y}_k + \mu\|\hat{s}_k\|^2} \\ &\leq \frac{\|\hat{y}_k\|^2 + \mu\hat{s}_k^T\hat{y}_k}{\hat{s}_k^T\hat{y}_k} + \mu\end{aligned}$$

$$\leq \frac{L^2 \xi_0^2 \|\hat{s}_k\|^2}{\hat{s}_k^T \hat{y}_k} + 2\mu$$

$$\leq \frac{L^2 \xi_0^2}{\upsilon} + 2\mu_{\max}.$$

The fourth inequality above is obtained from $\hat{y}_k = Z_k^T y_k, \|Z_k\| \leq \xi_0$ and Assumption 1 (ii). Because $\hat{B}_k(\mu)$ will be set to \hat{I} after a maximum of l updates, combining with (3.1) easy to get $\lambda_{\max}\left(\hat{B}_{k+1}(\mu)\right) \leq 1 + \frac{lL^2 \xi_0^2}{v} + 2l\mu_{\max} \triangleq \hat{\xi}_1.$

Let $\hat{P}_k(\mu) = \hat{B}_{k+1}^{-1}(\mu)$. According to (2.22) and some simple matrix operations, we have that $\hat{P}_k(\mu) = \hat{I}$ or

$$\hat{P}_{k}(\mu) = \left(\hat{I} - \frac{\hat{y}_{k}(\mu)\hat{s}_{k}^{T}}{\hat{s}_{k}^{T}\hat{y}_{k}(\mu)}\right)^{T}\hat{P}_{k-1}(\mu)\left(\hat{I} - \frac{\hat{y}_{k}(\mu)\hat{s}_{k}^{T}}{\hat{s}_{k}^{T}\hat{y}_{k}(\mu)}\right) + \frac{\hat{s}_{k}\hat{s}_{k}^{T}}{\hat{s}_{k}^{T}\hat{y}_{k}(\mu)}.$$
(3.2)

It is not difficult to that $\lambda_{\max}\left(\left(\hat{I} - \frac{\hat{y}_k(\mu)\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k(\mu)}\right)^T\left(\hat{I} - \frac{\hat{y}_k(\mu)\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k(\mu)}\right)\right) = \frac{\|\hat{y}_k(\mu)\|^2\|\hat{s}_k\|^2}{\left(\hat{s}_k^T\hat{y}_k(\mu)\right)^2}$. For any $\hat{z} \neq 0 \in \mathbb{R}^m$ and $\hat{P}_k(\mu)$ in (3.2), we have

$$\begin{split} \hat{z}^{T} \hat{P}_{k}(\mu) \hat{z} &= \hat{z}^{T} \left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right)^{T} \hat{P}_{k-1}(\mu) \left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right) \hat{z} + \frac{\left(\hat{s}_{k}^{T} \hat{z} \right)^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \\ &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \hat{z}^{T} \left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right)^{T} \left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right) \hat{z} + \frac{\left(\hat{s}_{k}^{T} \hat{z} \right)^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \\ &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \lambda_{\max} \left(\left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right)^{T} \left(\hat{I} - \frac{\hat{y}_{k}(\mu) \hat{s}_{k}^{T}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right) \right) \| \hat{z} \|^{2} + \frac{\left(\hat{s}_{k}^{T} \hat{z} \right)^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \\ &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \frac{\| \hat{y}_{k}(\mu) \|^{2} \| \hat{s}_{k} \|^{2}}{\left(\hat{s}_{k}^{T} \hat{y}_{k}(\mu) \right)^{2}} \| \hat{z} \|^{2} + \frac{\| \hat{s}_{k} \|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \| \hat{z} \|^{2}. \end{split}$$

The above inequality is divided by $\|\hat{z}\|^2$, and the resulting inequality is maximized, then we have

$$\begin{split} \lambda_{\max} \left(\hat{P}_{k}(\mu) \right) &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \frac{\|\hat{y}_{k}(\mu)\|^{2} \|\hat{s}_{k}\|^{2}}{\left(\hat{s}_{k}^{T} \hat{y}_{k}(\mu) \right)^{2}} + \frac{\|\hat{s}_{k}\|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \\ &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \left(\frac{\|\hat{y}_{k}(\mu)\|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}(\mu)} \right) + \frac{\|\hat{s}_{k}\|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}} \\ &\leq \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) \left(\frac{L^{2}\xi_{0}^{2}}{\upsilon} + 2\mu_{\max} \right) \frac{\|\hat{s}_{k}\|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}} + \frac{\|\hat{s}_{k}\|^{2}}{\hat{s}_{k}^{T} \hat{y}_{k}} \\ &\leq \left(\frac{L^{2}\xi_{0}^{2}}{\upsilon^{2}} + \frac{2\mu_{\max}}{\upsilon} \right) \lambda_{\max} \left(\hat{P}_{k-1}(\mu) \right) + \frac{1}{\upsilon}. \end{split}$$

The third inequality above is obtained from $\hat{y}_k = Z_k^T y_k$, $||Z_k|| \le \xi_0$ and Assumption 1 (ii). Because $\hat{P}_k(\mu)$ will be set to \hat{I} after a maximum of l updates, it is easy to know that there exists a constant $\hat{\xi}_2 > 0$ such that $\lambda_{\max}\left(\hat{B}_{k+1}^{-1}(\mu)\right) = \lambda_{\max}\left(\hat{P}_k(\mu)\right) \le \hat{\xi}_2$.

Since $\hat{B}_{k+1}^{-1}(\mu)$ is a positive definite and symmetric matrix, we have $\left\|\hat{B}_{k+1}^{-1}(\mu)\right\|_2 = \lambda_{\max}\left(\hat{B}_{k+1}^{-1}(\mu)\right) \leq \hat{\xi}_2$. As a result, using the equivalence property of matrix norm in a finite dimensional space, it follows that there exists a constant $\hat{\xi}_3 > 0$ such that $\left\|\hat{B}_{k+1}^{-1}(\mu)\right\| \leq \hat{\xi}_3$. The proof is completed.

$$\|P_k\| \le \gamma_0, \ g_{k+1}^T P_k g_{k+1} \ge \gamma_1 \|g_{k+1}\|^2, \ d_k^T P_k^{-1} d_k \ge \gamma_2 \|d_k\|^2,$$
(3.3)

where P_k^{-1} denotes the pseudoinverse of P_k .

Proof By (2.17), (2.27) and Lemma 1, we obtain that

$$\|P_k\| = \left\| Z_k \hat{B}_{k+1}^{-1}(\mu) Z_k^T \right\| = \left\| \hat{B}_{k+1}^{-1}(\mu) \right\| \le \hat{\xi}_3 \triangleq \gamma_0,$$

$$g_{k+1}^T P_k g_{k+1} = g_{k+1}^T Z_k \hat{B}_{k+1}^{-1}(\mu) Z_k^T g_{k+1}$$

= $\hat{g}_{k+1}^T \hat{B}_{k+1}^{-1}(\mu) \hat{g}_{k+1}$
 $\ge \lambda_{\min} \left(\hat{B}_{k+1}^{-1}(\mu) \right) \| \hat{g}_{k+1} \|^2$
 $\ge \frac{1}{\hat{\xi}_1} \left(1 - \tilde{\eta}_1^2 \right) \| g_{k+1} \|^2 \triangleq \gamma_1 \| g_{k+1} \|^2$

$$d_k^T P_k^{-1} d_k = d_k^T Z_k \hat{B}_{k+1}^{-1}(\mu) Z_k^T d_k = \hat{d}_k^T \hat{B}_{k+1}^{-1}(\mu) \hat{d}_k \ge \frac{1}{\hat{\xi}_2} \left\| \hat{d}_k \right\|^2 = \frac{1}{\hat{\xi}_2} \left\| d_k \right\|^2 \triangleq \gamma_2 \left\| d_k \right\|^2.$$

Therefore, we can get the conclusions. The proof is completed.

Subsequently, we provide some properties of the search directions produced by the algorithm RL_SMCG, which are crucial for the following convergence analysis.

Lemma 3 Suppose that Assumption 1 holds. Then, there exists a constant $c_1 > 0$ such that the search directions (2.12) and (2.26) are calculated by algorithm RL_SMCG satisfy the sufficient descent condition:

$$g_k^T d_k \le -\bar{c}_1 \|g_k\|^2. \tag{3.4}$$

Proof We divide the proof into the following two cases.

(i) SMCG iteration. Similar to the proof of Lemma 4.1 of [42], it is easy to have

$$g_k^T d_k \le -c_1 \|g_k\|^2,$$

where $c_1 = \min\left\{\frac{1}{2}, 1 - \bar{\xi}_3, \frac{2}{3\bar{\xi}_2}, \frac{1}{3\bar{\xi}_2}, \frac{2}{5\bar{\xi}_2}\right\}$.

(ii) RQN iteration. According to Lemma 2, we have

$$g_k^T d_k = -g_k^T P_{k-1} g_k \le -\gamma_1 \|g_k\|^2.$$

By setting $\bar{c}_1 = \min \{c_1, \gamma_1\}$, we can obtain (3.4). The proof is completed.

Lemma 4 Suppose that Assumption 1 holds. Then, there exists a constant $c_1 > 0$ such that the search directions (2.12) and (2.26) are calculated by algorithm RL_SMCG satisfy

$$\|d_k\| \le \bar{c}_2 \|g_k\|. \tag{3.5}$$

Proof We divide the proof into the following two cases.

(i) SMCG iteration. Referring to the proof procedure of Lemma 4.2 of [42], it is easy to get

$$\|d_k\| \le c_2 \|g_k\|$$

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

(ii) RQN iteration. According to Lemma 2, we obtain $||d_k|| = ||-P_{k-1}g_k|| \le \gamma_0 ||g_k||$.

By setting $\bar{c}_2 = \min \{c_2, \gamma_0\}$, we can obtain (3.5). The proof is completed.

The following lemmas are very critical for the convergence analysis of algorithm RL_SMCG.

Lemma 5 Suppose that Assumption 1 holds, and the sequence $\{x_k\}$ is generated by the algorithm RL_SMCG. Then,

If acceleration succeeds:

$$\bar{\eta}_k \alpha_k \ge \left(\frac{1-\sigma}{L}\right) \frac{\left|g_k^T d_k\right|}{\left\|d_k\right\|^2}.$$
(3.6)

If acceleration fails:

$$\alpha_k \ge \left(\frac{1-\sigma}{L}\right) \frac{\left|g_k^T d_k\right|}{\left\|d_k\right\|^2}.$$
(3.7)

Where σ are given by (2.43).

Proof We divide the proof into the following two cases.

(i) If acceleration succeeds:

From (2.43) and Assumptions 1 (ii), we obtain that

$$(\sigma - 1)g_k^T d_k \le g(x_k + \bar{\eta}_k \alpha_k d_k)^T d_k - g_k^T d_k = (g(x_k + \bar{\eta}_k \alpha_k d_k) - g_k)^T d_k \le L\bar{\eta}_k \alpha_k ||d_k||^2$$

which yields

$$\bar{\eta}_k \alpha_k \ge \left(\frac{\sigma - 1}{L}\right) \frac{g_k^T d_k}{\|d_k\|^2}$$

This means that (3.6) holds.

(ii) If acceleration fails:

Let $\bar{\eta}_k = 1$, and the rest of the proof procedure is the same as before.

Lemma 6 Suppose that Assumption 1 holds, and the sequence $\{x_k\}$ is generated by the algorithm RL_SMCG. Then, there holds that $f_k \leq C_k$ for each k.

Proof We divide the proof into the following two cases.

(i) If acceleration succeeds:

The new iterative update format is $x_{k+1} = x_k + \bar{\eta}_k \alpha_k d_k$, where $\bar{\eta}_k = -\frac{\bar{a}_k}{\bar{b}_k}$. Through (2.48), we have $f_{k+1} = f(x_k + \bar{\eta}_k \alpha_k d_k) \leq C_k + Q_{k+1} \delta_k \bar{\eta}_k \alpha_k g_k^T d_k$. Combining (2.44), $\delta_k > 0$, lemma 5 and the sufficiently descent property of the direction d_{k+1} , we have $f_{k+1} < C_k$. The remaining proof process refers to Lemma 5.1 in [42], we can obtain $f_{k+1} \leq C_{k+1}$, hence $f_k \leq C_k$ is established for each k.

(ii) If acceleration fails:

Let $\bar{\eta}_k = 1$, and the rest of the proof procedure is the same as before.

Theorem 1 Suppose that Assumption 1 holds, the sequence $\{x_k\}$ is generated by the algorithm RL-SMCG. Then,

$$\lim_{k \to \infty} \|g_k\| = 0. \tag{3.8}$$

Proof We divide the proof into the following two cases.

(i) If acceleration succeeds:

By Assumptions 1, lemmas 3 - 5 and the generalized nonmonotone Wolfe line search conditions (2.42) and (2.43), we get that

$$C_{k+1} \leq C_k + \delta_k \bar{\eta}_k \alpha_k g_k^T d_k$$

$$\leq C_k + \delta_{\min} \bar{\eta}_k \alpha_k g_k^T d_k$$

$$\leq C_k + \delta_{\min} \frac{1 - \sigma}{L} \frac{(g_k^T d_k)^2}{\|d_k\|^2}$$

$$\leq C_k + \frac{\delta_{\min}(1 - \sigma)\bar{c}_1^2}{L\bar{c}_2^2} \|g_k\|^2$$

$$= C_k + \beta \|g_k\|^2.$$
(3.9)

Where $\beta = \frac{\delta_{\min}(1-\sigma)\bar{c}_1^2}{L\bar{c}_2^2}$. Combined with (2.45), we have $C_1 \leq C_0$ that means that C_k is monotonically decreasing. According to lemma 6 and Assumption 1 (i), we know C_k is bounded from below. Then

$$\sum_{k=0}^\infty \beta \|g_k\|^2 < \infty,$$

therefore,

$$\lim_{k \to \infty} \|g(x_k)\| = 0$$

(ii) If acceleration fails:

Let $\bar{\eta}_k = 1$, and the rest of the proof procedure is the same as before.

4 Numerical Experiments

In this section, we compare the numerical performance of RL_SMCG with ASMCG_PR [32], CG_DESCENT(6.8) [15] and CGOPT(2.0) [27] for the 145 test problems from CUTEr library [10]. The codes of CG_DESCENT(6.8) [15] and CGOPT(2.0) [27] can be downloaded from http://users.clas.ufl.edu/hager/papers/Software and https://web.xidian.edu.cn/xdliuhongwei/en/paper.html or http://lsec.cc.ac.cn/ dyh/software.html, respectively.

In the numerical experiments, we set the parameters of RL_SMCG as: $\bar{\xi}_1 = 10^{-10}$, $\bar{\xi}_2 = 1.2 \times 10^4$, $\bar{\xi}_3 = 5 \times 10^{-5}$, $\bar{\xi}_4 = 10^{-4}$, $\bar{\xi}_5 = 0.08$, $\tilde{\eta}_0 = 10^{-9}$, $\tilde{\eta}_1 = 0.5$, $v = 5 \times 10^{-7}$, $m = \min\{n, 11\}$, $\sigma_1 = 0.1$, $\sigma_2 = 5$, $\sigma_3 = 0.85$, $\hat{\tau} = 1$, $\bar{\tau} = 0.225$, $\bar{c} = 0.1$, $\varsigma = 5 \times 10^{-5}$ ($n \le 11$), $\varsigma = 5 \times 10^{-6}$ (n > 11), $\bar{\varsigma} = 5 \times 10^{-3}$, $\tau_1 = 0.1$, $\tau_2 = 135$, $\delta_k = 0.0005$ and $\sigma = 0.9999$. CG_DESCENT(6.8) and CGOPT(2.0) take the default parameters in their codes but the stopping conditions. Note that the number of memory m for RL_SMCG is $\min\{n, 11\}$ while the number of memory for CG_DESCENT(6.8) is 11. All test methods in the experiment are terminated if $||g_k||_{\infty} \leq 10^{-6}$ is satisfied, and we set the number of iterations for all test algorithms to be no more than 200,000. In addition, all algorithms are running in Ubuntu 10.04 LTS.

We will show the performances of the test methods using the performance profiles introduced by Dolan and Moré [8]. In the following Figs. 1-12, " N_{iter} ", " N_f ", " N_g " and " T_{cpu} " represent the number of iterations, the number of function evaluations, the number of gradient evaluations and CPU time(s), respectively.

We divided the numerical experiments in three teams.



In the first set of numerical experiments, figures 1-4 illustrate the performance profiles of RL_SMCG and ASMCG_PR [32]. From Figs. 1, 2, 3 and 4, we can observe that RL_SMCG has a quite significant improvement over ASMCG_PR in terms of the number of iterations, the number of function evaluations, the number of gradient evaluations and CPU time. It indicates that the limited memory technique equipped in RL_SMCG indeed brings quite significant numerical improvements.

In the second set of numerical experiments, we give a comparison of the performance profiles of R-L_SMCG with CG_DESCENT(6.8) [15]. Regarding the number of iterations and the number of function evaluations in Fig. 5 and Fig. 6 respectively, we observe that RL_SMCG is a little better than CG_DESCENT(6.8) for the number of iterations and the number of function evaluations. As shown in Fig. 7, we can see that RL_SMCG is much better than CG_DESCENT(6.8) in terms of the number of gradient evaluations, because RL_SMCG outperforms for about 71.5% of the CUTEr test problems, while the percentage of software CG_DESCENT(6.8) is below 40%. It can be observe from Fig. 8 that RL_SMCG is faster than CG_DESCENT(6.8) in terms of CPU time. By Theorem 1, RL_SMCG is globally convergent with the generalized nonmonotone Wolfe line search, while CG_DESCENT (6.8) does not guarantee global convergence when using the rather efficient approximate Wolfe (AWolfe) line search. This means that RL_SMCG is superior to CG_DESCENT(6.8) for CUTEr library in theory and numerical performance.



In the third set of the numerical experiments, comparing the performance of RL_SMCG with CGOP-T(2.0) [27]. As shown in Figs. 9 and 10, we can take a look at RL_SMCG performs almost always better than CGOPT(2.0) in terms of the number of iterations and the number of function evaluations. Figures. 11 and 12 indicates that RL_SMCG outperforms CGOPT(2.0) in terms of the number of gradient evaluations and CPU time for the CUTEr library.

From the results of the above three numerical experiments, it is clear that the proposed algorithm RL_SMCG is quite effective.



5 Conclusions

In this paper, combined subspace minimization conjugate gradient method with limited memory technique, we presented a regularized limited memory subspace minimization conjugate gradient method, which contains two types of iteration. In the proposed algorithm, a modified regularized quasi-Newton method is given in small dimensional subspace to correct the orthogonality, and an improved initial step size selection strategy and some simple acceleration criteria are designed. Moreover, we establish the global convergence of the proposed algorithm by utilizing generalized nonmonotone Wolfe line search under some mild assumptions. Some numerical results suggest that our algorithm yields a tremendous improvement over the ASMCG_PR and outperforms the most up-to-date limited memory CG software packages CG_DESCENT (6.8) and CGOPT(2.0).

6 Declarations

6.1 Ethical Approval

Not Applicable

6.2 Availability of supporting data

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

6.3 Competing interests

The authors declare no competing interests.

6.4 Funding

This research was supported by the National Natural Science Foundation of China (No. 11901561), the Natural Science Foundation of Guizhou (No. ZK[2022]084) and the Natural Science Basic Research Program of Shaanxi (No. 2021JM-396).

6.5 Authors' contributions

Wumei Sun wrote the main manuscript text. Hongwei Liu and Zexian Liu reviewed and revised the manuscript.

6.6 Acknowledgments

The authors would like to thank the editor and the anonymous referees for their valuable suggestions and comments which have greatly improved the presentation of this paper.

References

- Andrei, N.: An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. Numer. Algor. 65, 859-874 (2014)
- 2. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer Anal. 8, 141-148 (1988)
- Dai, Y.H., Yuan, J.Y., Yuan, Y.X.: Modified two-point stepsize gradient methods for unconstrained optimization problems. Comput. Optim. Appl. 22(1), 103-109 (2002)
- Dai, Y.H.: Nonlinear Conjugate Gradient Methods. Wiley Encyclopedia of Operations Research and Management Science(2011). https://doi.org/10.1002/9780470400531.eorms0183
- Dai, Y.H., Kou, C.X.: A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM J. Optim. 23(1), 296-320 (2013)
- 6. Dai, Y.H., Kou, C.X.: A Barzilai-Borwein conjugate gradient method. Sci. China Math. 59(8), 1511-1524 (2016)
- Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. 10(1), 177-182 (1999)
- Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. 91, 201-213 (2002)
- 9. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Computer Journal. 7, 149-154 (1964)
- Gould, N.I.M., Orban, D., Toint, Ph.L: CUTEr and SifDec: A Constrained and Unconstrained Testing Environment, revisited. ACM Trans. Math. Softw. 29, 373-394 (2003)

- Gu, G.Z., Li, D.H., Qi, L.Q., Zhou, S.Z.: Descent directions of quasi-Newton methods for symmetric nonlinear equations. SIAM J. Numer. Anal. 40, 1763-1774 (2003)
- Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16(1), 170-192 (2005)
- 13. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. 2(1), 35-58 (2006)
- Hager, W.W., Zhang, H.: Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. ACM Trans. Math. Software. 32(1), 113-137 (2006)
- 15. Hager, W.W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. 23, 2150-2168 (2013)
- Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. 49, 409-436 (1952)
- Huang, S., Wan, Z., Chen, X.H.: A new nonmonotone line search technique for unconstrained optimization. Numer. Algor. 68(4), 671-689 (2015)
- Li, D.H., Fukushima, M.: A globally and superlinearly convergent Gauss-Newton-based BFGS methods for symmetric nonlinear equations. SIAM J. Numer. Anal. 37, 152-172 (1999)
- Li, D. H., Fukushima, M.: On the global convergence of BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. 11(4), 1054-1064 (2001)
- 20. Li, M., Liu, H.W., Liu, Z.X.: A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization. Numer Algor. **79**, 195-219 (2018)
- Li, Y.F., Liu, Z.X., Liu, H.W.: A subspace minimization conjugate gradient method based on conic model for unconstrained optimization. Computational and Applied Mathematics. 38(1), (2019)
- Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. 45, 503-528 (1989)
- Liu, T. W.: A regularized limited memory BFGS method for nonconvex unconstrained minimization. Numer. Algor. 65, 305-323 (2014)
- 24. Liu, Z.X., Liu, H.W.: An efficient gradient method with approximate optimal stepsize for large-scale unconstrained optimization. Numer. Algorithms **78**(1), 21-39 (2018)
- Liu, Z.X., Liu, H.W.: Several efficient gradient methods with approximate optimal stepsizes for large scale unconstrained optimization. J. Comput. Appl. Math. 328, 400-413 (2018)
- Liu, H.W., Liu, Z.X.: An efficient Barzilai-Borwein conjugate gradient method for unconstrained optimization. J. Optim. Theory Appl. 180, 879-906 (2019)
- Liu, Z.X., Liu, H.W., Dai, Y.H.: An improved Dai Kou conjugate gradient algorithm for unconstrained optimization. Comput. Optim. Appl. 75(1), 145-167 (2020)
- 28. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. 35, 773-782 (1980)
- 29. Nocedal, J., Wright, S.J.: Numerical Optimization. New York, Springer (1999)
- Polak, E., Ribière, G.: Note sur la convergence de méthodes de directions conjuguées. Rev. Franaise Informat. Rech. Opérationnelle. 3(16), 35-43 (1969)
- Polyak, B.T.: The conjugate gradient method in extremal problems. Ussr Comput. Math. Math. Phys. 9(4), 94-112 (1969)
- Sun, W., Liu, H., Liu, Z.: A Class of Accelerated Subspace Minimization Conjugate Gradient Methods. J. Optim. Theory Appl. 190(3), 811-840 (2021)
- Tarzangh, D.A., Peyghami, M.R.: A new regularized limited memory BFGS-type method based on modified secant conditions for unconstrained optimization problems. J. Global Optim. 63, 709-728 (2015)
- Tankaria, H., Sugimoto, S., Yamashita, N.: A regularized limited memory BFGS method for large-scale unconstrained optimization and its efficient implementations. Comput. Optim. Appl. 82, 61-88 (2022)
- 35. Ueda, K., Yamashita, N.: Convergence properties of the regularized newton method for the unconstrained nonconvex optimization. Appl. Math. Optim. **62**, 27-46 (2010)
- Wang, T., Liu, Z.X., Liu, H.W.: A new subspace minimization conjugate gradient method based on tensor model for unconstrained optimization. Int. J. Comput. Math. 96(10), 1924-1942 (2019)
- Yang, Y.T., Chen, Y.T. Lu, Y.L.: A subspace conjugate gradient algorithm for large-scale unconstrained optimization. Numer Algor. 76, 813-828 (2017)
- 38. Yuan, Y.X.: A modified BFGS algorithm for unconstrained optimization. IMA J. Numer. Anal. 11(3), 325-332 (1991)

- 39. Yuan, Y.X., Stoer, J.: A subspace study on conjugate gradient algorithms. Z. Angew. Math. Mech. 75(1), 69-77 (1995)
- 40. Yuan, Y. X., Sun, W. Y.: Theory and methods of optimization. Science Press of China (1999)
- Zhang, H., Hager, W.W.,: A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization. SIAM J. Optim. 14(4), 1043-1056 (2004)
- 42. Zhao, T., Liu, H.W., Liu, Z.X.: New subspace minimization conjugate gradient methods based on regularization model for unconstrained optimization. Numer. Algor. 87, 1501-1534 (2021)