RAIRO-Oper. Res. 58 (2024) 775–802
https://doi.org/10.1051/ro/2023161

RAIRO Operations Research
www.rairo-ro.org

AN ALTERNATIVE THREE-DIMENSIONAL SUBSPACE METHOD BASED ON CONIC MODEL FOR UNCONSTRAINED OPTIMIZATION

Guoxin Wang[1], Mingyang Pei[1], Zengxin Wei[2] and Shengwei Yao[3,*]

Abstract. In this paper, a three-dimensional subspace conjugate gradient method is proposed, in which the search direction is generated by minimizing the approximation model of the objective function in a three-dimensional subspace. The approximation model is not unique and is alternative between quadratic model and conic model by the specific criterions. The strategy of initial stepsize and nonmonotone line search are adopted, and the global convergence of the presented algorithm is established under mild assumptions. In numerical experiments, we use a collection of 80 unconstrained optimization test problems to show the competitive performance of the presented method.

Mathematics Subject Classification. 90C30, 90C26.

Received November 2, 2022. Accepted October 3, 2023.

1. Introduction

Considering the following unconstrained problem:

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

with an initial point x_0, the following iterative formula is often used to solve (1.1),

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

where x_k is the k-th iteration point, $\alpha_k \in \mathbb{R}$ is the stepsize determined by a line search procedure, and d_k is the search direction acquired by specific ways.

Conjugate gradient methods are one of the common methods for unconstrained optimization problems, of which the search direction is computed as

$$d_{k+1} = -g_{k+1} + \beta_k d_k,\ k = 0, 1, \ldots, \tag{1.3}$$

where $d_0 = -g_0$, $g_{k+1} = \nabla f(x_{k+1})$ and $\beta_k \in \mathbb{R}$ is a scalar called the conjugate gradient parameter. Corresponding to different choices for the parameter β_k, various nonlinear conjugate gradient methods have been proposed. Some classical CG methods include HS (Hestenes and Stiefel [22]), FR (Fletcher and Reeves [20]), PRP (Polak *et al.* [30]), CD (Fletcher [19]), LS (Liu and Storey [25]) and DY (Dai and Yuan [11]).

Keywords. Unconstrained optimization, subspace, conic model, global convergence.

[1] School of Mathematics and Information Science, Guangxi University, Nanning 530004, P.R. China.

[2] Xiangsihu College of GuangXi University for Nationalities, Nanning 530000, P.R. China.

[3] College of Mathematics and Statistics, Nanning Normal University, Nanning 530001, P.R. China.

*Corresponding author: ysw@gxu.edu.cn

As for the stepsize α_k used in (1.2), it depicts the length of current point x_k increasing along with the generated search direction d_k, and is usually determined by a procedure known as line search. The largest reduction of the function value is achieved when the exact line search is exploited, where

$$\alpha_k = \arg\min_{\alpha \geq 0} f(x_k + \alpha d_k),$$

and such a stepsize is called the exact stepsize. However, the high cost and difficulty for computing the exact stepsize make it rarely used in optimization algorithms. Instead, an inexact line search is often used. One of the most used inexact line search is the so-called standard Wolfe line search [37,38]:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \tag{1.4}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{1.5}$$

where $0 < \delta < \sigma < 1$. Obviously, it is a monotone procedure that seeks for a suitable α_k making the function value decrease to some extent. Zhang and Hager [45] proposed a nonmonotone version (ZH line search) that modifies condition (1.4) to

$$f(x_k + \alpha_k d_k) \leq C_k + \delta \alpha_k g_k^T d_k, \tag{1.6}$$

where $C_0 = f(x_0)$, $Q_0 = 1$, C_{k+1} and Q_{k+1} are updated by

$$C_{k+1} = \frac{\eta_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}, \quad Q_{k+1} = \eta_k Q_k + 1, \tag{1.7}$$

where $\eta_k \in [\eta_{\min}, \eta_{\max}]$ and $0 \leq \eta_{\min} \leq \eta_{\max} \leq 1$. The choice of η_k controls the degree of nonmonotonicity. Such a line search can not only overcome some drawbacks in monotone line search, but is particularly efficient for unconstrained problems in numerical experiments [45].

Subspace technique is one of the effective means for solving large-scale optimization problems, which is getting more and more attention. Yuan reviewed various subspace techniques that have been used in constructing numerical methods for solving nonlinear optimization problems in [42,43]. Moreover, the combination between subspace technique and conjugate gradient method has been extensively studied. The earliest research can see [44], Yuan and Stoer computed the search direction d_{k+1} by minimizing the approximation quadratic model in the two dimensional subspace spanned by g_{k+1} and s_k, namely $\Omega_{k+1} = Span\{g_{k+1}, s_k\}$ where $s_k = x_{k+1} - x_k$, and proposed the subspace minimization conjugate gradient method (SMCG), in which d_{k+1} is formed by

$$d_{k+1} = t g_{k+1} + \mu s_k, \tag{1.8}$$

where t and μ are undetermined parameters. Based on the above idea, Andrei [3] extended the subspace to $\Omega_{k+1} = Span\{g_{k+1}, s_k, y_k\}$ and exploited the acceleration scheme, finally presented a three-term conjugate gradient method (TTS), in which

$$d_{k+1} = -g_{k+1} + \mu s_k + \nu y_k,$$

and $y_k = g_{k+1} - g_k$, μ, ν are also scalar parameters. Inspired by Andrei, Yang *et al.* [39] changed the subspace into $\Omega_{k+1} = Span\{g_{k+1}, s_k, s_{k-1}\}$, and put forward the subspace three-term conjugate gradient method (STT). For the same subspace, Li *et al.* [23] added more parameters to the computation of search direction so that

$$d_{k+1} = t g_{k+1} + \mu s_k + \nu s_{k-1},$$

and adopted the strategy of initial stepsize as well as the nonmonotone line search, eventually proposed the subspace minimization conjugate gradient method with nonmonotone line search (SMCG_NLS).

On the other hand, Dai and Kou [13] also focused on the analysis of Yuan and Stoer [44], but they paid more attention to the estimate of the parameter $\rho_{k+1} = g_{k+1}^T B_{k+1} g_{k+1}$ during the calculation of d_{k+1}. They combined the Barzilai-Borwein [7] idea and provided some efficient Barzilai-Borwein conjugate gradient methods (BBCG).

It is remarkable that the idea of BBCG to estimate ρ_{k+1} is employed in this paper. Motivated by SMCG and BBCG, Liu and Liu [27] proposed a new Barzilai-Borwein conjugate gradient method (SMCG_BB) with a new strategy for the choice of initial stepsize and a nonmonotone generalized Wolfe line search.

It is noteworthy that all of the above mentioned subspace minimization conjugate gradient methods obtained the search direction by minimizing the approximate quadratic model of objective function in the presented subspace. However, Sun [34] and Sun and Yuan [35] have pointed that when the current iterative point is not close to the minimizer, the quadratic model may lead to a poor prediction of the minimizer if the objective function possesses strong non-quadratic behaviour. Besides this, a quadratic model does not take into account more information instead of the gradient value in the current iteration, which means that it does not have enough degrees freedom for incorporating all of the information in the iterative procedure.

Thus, the research for approximate nonquadratic model is of the essence. Up to now, many nonquadratic models have been applied to optimization problems, such as conic model, tensor model and regularization model. The conic model can be incorporated in more function information than quadratic model, and its application in unconstrained optimization was first studied by Davidon [16]. A typical conic model for unconstrained optimization is

$$\phi_{k+1}(s) = \frac{g_{k+1}^T s}{1 + b_{k+1}^T s} + \frac{1}{2}\frac{s^T B_{k+1} s}{(1 + b_{k+1}^T s)^2},$$

which is an approximation to $f(x_k + s) - f(x_k)$, and B_{k+1} is a symmetric positive definite matrix approximating to the Hessian of $f(x)$ at x_{k+1} satisfying the secant equation $B_{k+1}s_k = y_k$. The vector b_{k+1} is normally called the horizontal vector satisfying $1 + b_{k+1}^T s > 0$. Such a conic model has been investigated by many scholars. Sorensen [33] discussed a class of conic methods called "optimization by collinear scaling" for unconstrained optimization and shown that a particular member of this algorithm class has a Q-superlinear convergence. Ariyawansa [5] modified the procedure of [33] and established the duality between the collinear scaling DFP and BFGS methods. Sheng [32] further discussed the interpolation properties of conic model method. Di and Sun [17] proposed a trust region method for conic models to solve unconstrained optimization problems. The trust region methods based on conic model have brought about a great number of research.

Li *et al.* [24] paid attention to the combination of subspace method and conic model. They considered the following conic approximation model:

$$\phi_{k+1}(d) = \frac{g_{k+1}^T d}{1 + b_{k+1}^T d} + \frac{1}{2}\frac{d^T B_{k+1} d}{(1 + b_{k+1}^T d)^2} \tag{1.9}$$

where

$$\begin{aligned} b_{k+1} &= -\frac{1 - \gamma_{k+1}}{\gamma_{k+1} g_{k+1}^T s_k} g_{k+1}, \\ \gamma_{k+1} &= \frac{-g_k^T s_k}{\sqrt{\Delta_{k+1}} + f_k - f_{k-1}}, \\ \Delta_{k+1} &= (f_k - f_{k+1})^2 - (g_{k+1}^T s_k)(g_k^T s_k). \end{aligned}$$

Note that f_k denotes $f(x_k)$. By minimizing the above conic model in the two-dimensional subspace $\Omega_{k+1} = Span\{g_{k+1}, s_k\}$, they developed a subspace minimization conjugate method based on the conic model (SMCG_Conic). Sun *et al.* [36] extended the subspace to $\Omega_{k+1} = Span\{g_{k+1}, s_k, s_{k-1}\}$ and presented a three-dimensional subspace minimization conjugate gradient method based on conic model (CONIC_CG3).

Inspired by [3] and above mentioned works, we come up with the question whether we can extend the two-dimensional subspace in [24] to three-dimensional $\Omega_{k+1} = Span\{g_{k+1}, s_k, y_k\}$. And does the generated algorithm has the property of global convergence and competitive numerical performance? Therefore, this paper investigates a three-dimensional subspace method based on the conic model (1.9). Furthermore, some schemes helpful for convergence are taken into account.

This paper is organized as follows: in Section 2, the search directions on the subspace Ω_{k+1} based on two different models are derived, and the criteria for how to choose the approximate models and search direction are presented. In Section 3, the schemes of initial stepsize and nonmonotone line search are presented, and the generated algorithm will be detailed. In Section 4, we give the proofs for some important lemmas of the search direction and the convergence performance of the generated algorithm under suitable assumptions and conditions. In Section 5, we compare the numerical results of our algorithm with those of another two methods.

2. The search direction based on different models

The main content of this section is to construct the formula of search direction under different situations and the corresponding criteria. The concrete approach is to minimize the approximation model in the subspace $\Omega_{k+1} = Span\{g_{k+1}, s_k, y_k\}$, hence how to choose the appropriate approximation model is crucially important.

Here we refer to the analysis of Yuan [40], in which a quantity u_k is defined by

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

which shows the extent of how the objective function $f(x)$ is close to a quadratic on the line segment between x_k and x_{k+1}. Dai *et al.* [15] indicate that if the following condition

$$u_k \leq c_1 \ or \ \max\{u_k, u_{k-1}\} \leq c_2 \tag{2.1}$$

holds, where $0 < c_1 < c_2$ are two small constants, then they believe that $f(x)$ is very close to a quadratic on the line segment between x_k and x_{k+1}. The utilization of such a quantity can be referred to [26,27]. In this paper, if the above condition (2.1) is satisfied, then the choice of the quadratic approximation model is preferable; otherwise, the conic model is more suitable.

Since we figure out the criterion for choosing the approximation model, the following is to establish the formula of search direction based on the specific model and subspace. Furthermore, the situations for different dimensions of Ω_{k+1} ranging from 1 to 3 are taken into account.

2.1. Conic model

In this subsection, we consider the subproblem

$$\min_{d \in \Omega_{k+1}} \phi_{k+1}(d), \tag{2.2}$$

where $\phi_{k+1}(d)$ is the same as (1.9).

Three different dimensions under the subspace $\Omega_{k+1} = Span\{g_{k+1}, s_k, y_k\}$ will be discussed.

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

Under this situation, the search direction is computed by

$$d_{k+1} = t g_{k+1} + \mu s_k + \nu y_k. \tag{2.3}$$

By substituting (2.3) into (2.2) and using the secant equation, the problem (2.2) turns into

$$\min_{(d)} \phi_{k+1}(t,\mu,\nu) =$$

$$\frac{\begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}}{1 + \begin{pmatrix} b_{k+1}^T g_{k+1} \\ b_{k+1}^T s_k \\ b_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}} + \frac{1}{2} \frac{\begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k & \omega_k \\ g_{k+1}^T y_k & s_k^T y_k & y_k^T y_k \\ \omega_k & y_k^T y_k & \tau_k \end{pmatrix} \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}}{\left[1 + \begin{pmatrix} b_{k+1}^T g_{k+1} \\ b_{k+1}^T s_k \\ b_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}\right]^2}, \tag{2.4}$$

where $\rho_{k+1} = g_{k+1}^T B_{k+1} g_{k+1},\ \omega_k = g_{k+1}^T B_{k+1} y_k,\ \tau_k = y_k^T B_{k+1} y_k$, and $\|.\|$ denotes the Euclidean norm. We set

$$a = \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix}, c = \begin{pmatrix} b_{k+1}^T g_{k+1} \\ b_{k+1}^T s_k \\ b_{k+1}^T y_k \end{pmatrix}, u = \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix},$$

$$A_{k+1} = \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k & \omega_k \\ g_{k+1}^T y_k & s_k^T y_k & y_k^T y_k \\ \omega_k & y_k^T y_k & \tau_k \end{pmatrix},$$

thus (2.4) turns into

$$\phi_{k+1}(u) = \frac{a^T u}{1 + c^T u} + \frac{1}{2} \frac{u^T A_{k+1} u}{(1 + c^T u)^2}. \tag{2.5}$$

To minimize (2.5), we derive its first derivative and seek the solution of

$$\nabla \phi_{k+1}(u) = \frac{1}{1 + c^T u} \left(I - \frac{c u^T}{1 + c^T u} \right) \left(a + \frac{A_{k+1} u}{1 + c^T u} \right) = 0.$$

Obviously that $I - \frac{cu^T}{1+c^T u}$ is invertible, so the problem is reduced to $a + \frac{A_{k+1} u}{1+c^T u} = 0$, then we can easily acquire the minimizer of (2.5)

$$u_{k+1} = \frac{-A_{k+1}^{-1} a}{1 + c^T A_{k+1}^{-1} a}, \tag{2.6}$$

when A_{k+1} is positive definite and $1 + c^T A_{k+1}^{-1} a \neq 0$. Moreover, by using the relationship $A_{k+1}^{-1} = \frac{A_{k+1}^*}{|A_{k+1}|}$, where

$$A_{k+1}^* = \begin{pmatrix} X & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & Y \end{pmatrix}, \tag{2.7}$$

is the adjoint matrix of A_{k+1}, and $|A_{k+1}| = \rho_{k+1} X + \theta_1 g_{k+1}^T y_k + \theta_2 \omega_k$, in which

$$\begin{aligned} X &= (s_k^T y_k)\tau_k - (y_k^T y_k)^2, \\ \theta_1 &= (y_k^T y_k)\omega_k - (g_{k+1}^T y_k)\tau_k, \\ \theta_2 &= (g_{k+1}^T y_k)(y_k^T y_k) - (s_k^T y_k)\omega_k, \\ \theta &= \rho_{k+1}\tau_k - \omega_k^2, \\ \theta_3 &= (g_{k+1}^T y_k)\omega_k - \rho_{k+1}(y_k^T y_k), \\ Y &= \rho_{k+1}(s_k^T y_k) - (g_{k+1}^T y_k)^2, \end{aligned}$$

we finally obtain the minimizer of (2.4)

$$u_{k+1} = \begin{pmatrix} t_{k+1} \\ \mu_{k+1} \\ \nu_{k+1} \end{pmatrix} = -\frac{1}{D_{k+1}} \begin{pmatrix} X & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & Y \end{pmatrix} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix} = \frac{1}{D_{k+1}} \begin{pmatrix} -q_1 \\ -q_2 \\ -q_3 \end{pmatrix}, \tag{2.8}$$

where

$$\begin{aligned} D_{k+1} &= |A_{k+1}| + q_1 b_{k+1}^T g_{k+1} + q_2 b_{k+1}^T s_k + q_3 b_{k+1}^T y_k, \\ q_1 &= X \|g_{k+1}\|^2 + \theta_1 g_{k+1}^T s_k + \theta_2 g_{k+1}^T y_k, \\ q_2 &= \theta_1 \|g_{k+1}\|^2 + \theta g_{k+1}^T s_k + \theta_3 g_{k+1}^T y_k, \\ q_3 &= \theta_2 \|g_{k+1}\|^2 + \theta_3 g_{k+1}^T s_k + Y g_{k+1}^T y_k. \end{aligned}$$

So the solution of (2.2) in three dimensional subspace $\Omega_{k+1} = Span\{g_{k+1}, s_k, y_k\}$ is

$$d_{k+1} = \left(g_{k+1}\ s_k\ y_k\right) u_{k+1}.$$

However, there are three quantities in A_{k+1} that need to be estimated appropriately in order to avoid matrix-vector multiplication and improve efficiency, *i.e.* ρ_{k+1}, ω_k and τ_k.

As a matter of fact, the estimate for ρ_{k+1} is an essential procedure in the subspace method. Yuan and Stoer [44] proposed two ways to calculate such quantities containing B_{k+1}, one of which is to obtain B_{k+1} by using the scaled memoryless BFGS formula. The approach of Dai and Kou [13] is to combine the Barzilai-Borwein [7] idea by approximating the Hessian by $(1/\alpha_{k+1}^{BB_1})I$ or $(1/\alpha_{k+1}^{BB_2})I$, where

$$\alpha_{k+1}^{BB_1} = \frac{\|s_k\|^2}{s_k^T y_k}, \quad \alpha_{k+1}^{BB_2} = \frac{s_k^T y_k}{\|y_k\|^2}\cdot$$

Here we adopt the idea of Li *et al.* [23], because it can guarantee some good properties.

Firstly, according to the analysis about (2.6), the positive definitiveness of A_{k+1} is an essential condition requiring $|A_{k+1}| > 0$, it follows

$$\rho_{k+1} > \frac{-\theta_1 g_{k+1}^T y_k - \theta_2 \omega_k}{X}\cdot \tag{2.9}$$

By setting $X = m_k s_k^T y_k \tau_k$ with $m_k \triangleq 1 - \frac{(y_k^T y_k)^2}{s_k^T y_k \tau_k}$, we have

$$\rho_{k+1} > \left[\frac{(g_{k+1}^T y_k)^2}{s_k^T y_k} + \frac{\omega_k^2}{\tau_k} - 2\frac{g_{k+1}^T y_k \omega_k y_k^T y_k}{s_k^T y_k \tau_k}\right] / m_k \triangleq n_k, \tag{2.10}$$

if and only if m_k is positive, which can be guaranteed by (2.15). Note that we define the right-hand side of (2.10) as n_k. In addition, the positive definitiveness of A_{k+1} also requires that its first and second order leading principal minors are positive, and it means

$$\rho_{k+1} > \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k}\cdot \tag{2.11}$$

Secondly, $D_{k+1} > 0$ is also a necessary condition to keep the sufficient descent property, which follows

$$\rho_{k+1} > \frac{S_k}{s_k^T y_k \tau_k} / M_k, \tag{2.12}$$

if $M_k > 0$, where

$$\begin{aligned} S_k = -\theta_1 g_{k+1}^T y_k - \theta_2 \omega_k &- b_{k+1}^T g_{k+1} (X\|g_{k+1}\|^2 + \theta_1 g_{k+1}^T s_k + \theta_2 g_{k+1}^T y_k) \\ &- b_{k+1}^T s_k (\theta_1 \|g_{k+1}\|^2 - \omega_k^2 g_{k+1}^T s_k + \omega_k (g_{k+1}^T y_k)^2) \\ &- b_{k+1}^T y_k (\theta_2 \|g_{k+1}\|^2 - (g_{k+1}^T y_k)^3 + \omega_k g_{k+1}^T y_k g_{k+1}^T s_k), \end{aligned}$$

and

$$M_k = 1 - \frac{(y_k^T y_k)^2}{s_k^T y_k \tau_k} + \frac{1 - \gamma_{k+1}}{\gamma_{k+1}} \left[2\frac{g_{k+1}^T y_k y_k^T y_k}{s_k^T y_k \tau_k} - \frac{g_{k+1}^T s_k}{s_k^T y_k} - \frac{(g_{k+1}^T y_k)^2}{g_{k+1}^T s_k \tau_k}\right]\cdot$$

Likewise, we define the right-hand side of (2.12) as N_k, *i.e.* $N_k \triangleq \frac{S_k}{s_k^T y_k \tau_k} / M_k$.

After the above discusssion, we can estimate ρ_{k+1} as follows,

$$\rho_{k+1} = \zeta_k \max\{K, N_k, n_k\}, \tag{2.13}$$

where $K = K_1\|g_{k+1}\|^2$,

$$K_1 = \max\left\{\frac{\|y_k\|^2}{s_k^T y_k}, |\frac{1-\gamma_{k+1}}{\gamma_{k+1}}|\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|}\right\},$$

and

$$\zeta_k = \begin{cases}\max\{0.9\zeta_{k-1}, 1.2\}, & \text{if } \alpha_k > 1.0,\\ \min\{1.1\zeta_{k-1}, 1.75\}, & \text{otherwise,}\end{cases}$$

with $\zeta_0 = 1.5$, it is obvious that $\zeta_k \in [1.2, 1.75]$.

Observe the composition of (2.13), you can find that such a formula of ρ_{k+1} is more than an estimate, but can guarantee (2.10), (2.11) and (2.12). Besides, it makes d_{k+1} a descent direction, which will be proved in Section 4.

Then we estimate $\omega_k = g_{k+1}^T B_{k+1} y_k$ and $\tau_k = y_k^T B_{k+1} y_k$. For ω_k, we utilize the memoryless BFGS formula to get B_{k+1} so that

$$\begin{aligned}\omega_k &= g_{k+1}^T\left(I + \frac{y_k y_k^T}{s_k^T y_k} - \frac{s_k s_k^T}{s_k^T s_k}\right) y_k\\ &= g_{k+1}^T y_k + \frac{g_{k+1}^T y_k y_k^T y_k}{s_k^T y_k} - \frac{g_{k+1}^T s_k s_k^T y_k}{s_k^T s_k}.\end{aligned} \tag{2.14}$$

Then for τ_k, we combine the idea of [13] and [23], and estimate τ_k by

$$\tau_k = \zeta_k \frac{\|y_k\|^2}{s_k^T y_k} y_k^T y_k. \tag{2.15}$$

Before computing d_{k+1} by (2.3) and (2.8), we should verify the following conditions:

$$\Delta_{k+1} \geq 0, \tag{2.16}$$

$$M_k \geq \rho_0, \tag{2.17}$$

$$\xi_1 \leq \frac{s_k^T y_k}{\|s_k\|^2} \leq \frac{\|y_k\|^2}{s_k^T y_k} \leq \xi_2, \tag{2.18}$$

$$\left|\frac{1-\gamma_{k+1}}{\gamma_{k+1}}\right| \frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} \leq \xi_3, \tag{2.19}$$

$$\frac{|A_{k+1}|}{s_k^T y_k \tau_k \rho_{k+1}} \geq \xi_5, \tag{2.20}$$

where $\rho_0 \in (0,1)$ and $\xi_1, \xi_2, \xi_3, \xi_5$ are positive constants. (2.16) and (2.17) are fundamental premise of the conic model (1.5) and relation (2.12), respectively. On the basis of the Barzilai-Borwein [7] idea, (2.18) might indicate the suitable condition numbers of the approximation Hessian matrix. (2.19) is vital to guarantee the descent property of the search direction. As for (2.20), obviously it makes A_{k+1} more positive definite, and is also helpful for establishing the sufficient descent property of the search direction.

Therefore, if (2.16)–(2.20) hold, we compute the search direction by (2.3) and (2.8).

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

Li *et al.* [24] have made a deep study of the subspace conjugate gradient method based on conic model in this case, here we refer to their works. When $dim(\Omega_{k+1}) = 2$, the search direction is formed by

$$d_{k+1} = t g_{k+1} + \mu s_k,$$

which is the same as (1.8). The formula of t and μ is

$$\begin{pmatrix} t_{k+1} \\ \mu_{k+1} \end{pmatrix} = \frac{1}{\bar{D}_{k+1}} \begin{pmatrix} -c_1 \\ -c_2 \end{pmatrix}, \tag{2.21}$$

which is the same as (13) in [24]. Whether the search direction is computed by (1.8) and (2.21) depend on whether the following conditions hold or not,

$$\Delta_{k+1} \geq 0, \tag{2.22}$$

$$\bar{m}_k \geq \bar{\rho}_0, \tag{2.23}$$

$$\xi_1 \leq \frac{s_k^T y_k}{\|s_k\|^2} \leq \frac{\|y_k\|^2}{s_k^T y_k} \leq \xi_2, \tag{2.24}$$

$$\left| \frac{1-\gamma_{k+1}}{\gamma_{k+1}} \right| \frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} \leq \xi_3, \text{ if } \frac{1-\gamma_{k+1}}{\gamma_{k+1} g_{k+1}^T s_k} < 0, \tag{2.25}$$

$$\frac{\|g_{k+1}\|^2 \|y_k\| \|s_k\|}{(g_{k+1}^T s_k)^2} \leq \xi_4, \text{ if } \frac{1-\gamma_{k+1}}{\gamma_{k+1} g_{k+1}^T s_k} \geq 0, \tag{2.26}$$

where $\bar{\rho}_0 \in (0,1)$, ξ_4 is a positive constant, $\xi_1,\ \xi_2,\ \xi_3$ are identical to those in (2.18) and (2.19). If (2.16)–(2.20) do not all hold but (2.22)–(2.26) hold, we compute the search direction by (1.8) and (2.21).

However, if any of the conditions (2.22)–(2.26) fails but (2.27) and (2.28) hold,

$$\vartheta_1 \leq \frac{s_k^T y_k}{\|s_k\|^2}, \tag{2.27}$$

$$\frac{|g_{k+1}^T y_k g_{k+1}^T d_k|}{d_k^T y_k \|g_{k+1}\|^2} \leq \vartheta_5, \tag{2.28}$$

where $\vartheta_5 \in [0,1)$ and ϑ_1 is a positive constant, we consider the HS direction,

$$d_{k+1} = -g_{k+1} + \beta_k^{HS} d_k. \tag{2.29}$$

Therefore, there are two choices of search direction when $dim(\Omega_{k+1}) = 2$: one is to compute d_{k+1} by (1.8) and (2.21) if (2.22)–(2.26) hold; the other is to compute d_{k+1} by (2.29) if (2.27) and (2.28) hold. Otherwise, we use the negative gradient direction $-g_{k+1}$ as our search direction, which means $dim(\Omega_{k+1}) = 1$.

2.2. Quadratic model

When (2.1) is satisfied, we consider the subproblem based on quadratic approximation model as follows:

$$\min_{d \in \Omega_{k+1}} \psi_{k+1}(d) = g_{k+1}^T d + \frac{1}{2} d^T B_{k+1} d. \tag{2.30}$$

There also exist three situations.

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

Similar to the analysis in the first situation of Subsection 2.1, we substitute (2.3) into (2.30), then we have

$$\min_{(t,\mu,\nu)} \psi_{k+1}(t,\mu,\nu) = \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix} + \frac{1}{2} \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k & \omega_k \\ g_{k+1}^T y_k & s_k^T y_k & y_k^T y_k \\ \omega_k & y_k^T y_k & \tau_k \end{pmatrix} \begin{pmatrix} t \\ \mu \\ \nu \end{pmatrix}. \tag{2.31}$$

Obviously, if A_{k+1} is positive definite, we can easily obtain the solution of (2.31) as follows,

$$\begin{aligned}\begin{pmatrix} t_{k+1} \\ \mu_{k+1} \\ \nu_{k+1} \end{pmatrix} &= -A_{k+1}^{-1}\begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix} \\ &= -\frac{1}{|A_{k+1}|}\begin{pmatrix} X & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & Y \end{pmatrix}\begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix} \\ &= \frac{1}{|A_{k+1}|}\begin{pmatrix} -q_1 \\ -q_2 \\ -q_3 \end{pmatrix},\end{aligned} \tag{2.32}$$

in which $q_1,\ q_2,\ q_3$ are the same as those in (2.8).

In Subsection 2.1, we adopt a special way to estimate ρ_{k+1} because it can guarantee some important properties while the conic approximation model is employed, but this method requires much computation as well as comparison. BBCG combining the Barzilai-Borwein idea is one of the efficient SMCG methods based on quadratic model, especially BBCG3. Therefore, we take the approach similar to BBCG3 in the case that quadratic model is selected. Actually, this approach has be mentioned before, that is the way we estimate τ_k in (2.15).

In (2.15), we use $(\|y_k\|^2/s_k^T y_k)I$ to approximate the Hessian matrix and combine it with an adaptive parameter ζ_k. We do the same thing to ρ_{k+1}, hence we have the following formula,

$$\rho_{k+1} = \zeta_k \frac{\|y_k\|^2}{s_k^T y_k}\|g_{k+1}\|^2. \tag{2.33}$$

Likewise, it is reasonable to compute ω_k by the same way due to its similar formula to ρ_{k+1}, so we have

$$\omega_k = \zeta_k \frac{\|y_k\|^2}{s_k^T y_k} g_{k+1}^T y_k.$$

Obviously, such a choice of ρ_{k+1} satisfies both $\rho_{k+1} > 0$ and

$$\rho_{k+1} > \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k},$$

which means the first and second order leading principal minors of A_{k+1} are both positive, so A_{k+1} is positive definite as long as its determinant is positive. We notice that $|A_{k+1}| = \rho_{k+1}X + \theta_1 g_{k+1}^T y_k + \theta_2\omega_k$, then through calculation and simplification, we have

$$\begin{aligned}|A_{k+1}| &= \rho_{k+1}s_k^T y_k\tau_k - \rho_{k+1}\|y_k\|^4 + 2g_{k+1}^T y_k\|y_k\|^2\omega_k - (g_{k+1}^T y_k)^2\tau_k - s_k^T y_k\omega_k^2 \\ &= \rho_{k+1}(\zeta_k - 1)\|y_k\|^4 + (\zeta_k - \zeta_k^2)\frac{(g_{k+1}^T y_k)^2\|y_k\|^4}{s_k^T y_k} \\ &= \zeta_k(\zeta_k - 1)\frac{\|y_k\|^4}{s_k^T y_k}[\|g_{k+1}\|^2\|y_k\|^2 - (g_{k+1}^T y_k)^2] \\ &= \zeta_k(\zeta_k - 1)\frac{\|y_k\|^4}{s_k^T y_k}\left[1 - \frac{(g_{k+1}^T y_k)^2}{\|g_{k+1}\|^2\|y_k\|^2}\right]\|g_{k+1}\|^2\|y_k\|^2.\end{aligned} \tag{2.34}$$

From the above quality, we can know that A_{k+1} is positive definite if and only if g_{k+1} and y_k are linearly independent, which can be described as

$$\frac{(g_{k+1}^T y_k)^2}{\|g_{k+1}\|^2\|y_k\|^2} < 1.$$

Therefore, when the following conditions hold,

$$\vartheta_1 \leq \frac{s_k^T y_k}{\|s_k\|^2} \leq \frac{\|y_k\|^2}{s_k^T y_k} \leq \vartheta_2, \tag{2.35}$$

$$1 - \frac{(g_{k+1}^T y_k)^2}{\|g_{k+1}\|^2 \|y_k\|^2} \geq \vartheta_3, \tag{2.36}$$

$$\frac{\|s_k\|^2}{\|g_{k+1}\|^2} \geq \vartheta_4, \tag{2.37}$$

we compute d_{k+1} by (2.3) and (2.32), where $\vartheta_3 \in (0,1)$ and ϑ_2, ϑ_4 are positive constants. (2.35) plays the same role as (2.24). The relation (2.36) is essential to the positive definiteness of A_{k+1} and the sufficient descent property of d_{k+1}. (2.37) means that s_k has an adaptive lower bound such that the objective function can descend along s_k.

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

In this situation, due to the better numerical performance based on the subspace $span\{g_{k+1}, s_k\}$, we only consider the search direction formed by (1.8), then we add it into (2.30) and can obtain the prediction subproblem

$$\min_{(t,\mu)} \psi_{k+1}(t,\mu) = \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \end{pmatrix}^T \begin{pmatrix} t \\ \mu \end{pmatrix} + \frac{1}{2} \begin{pmatrix} t \\ \mu \end{pmatrix}^T \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k \\ g_{k+1}^T y_k & s_k^T y_k \end{pmatrix} \begin{pmatrix} t \\ \mu \end{pmatrix}. \tag{2.38}$$

Here we still compute ρ_{k+1} by (2.33), which leads to

$$|\bar{A}_{k+1}| = \rho_{k+1} s_k^T y_k - (g_{k+1}^T y_k)^2 > 0,$$

where

$$\bar{A}_{k+1} = \begin{pmatrix} \rho_{k+1} & g_{k+1}^T y_k \\ g_{k+1}^T y_k & s_k^T y_k \end{pmatrix},$$

so the problem (2.38) has unique solution; that is

$$\begin{pmatrix} t_k \\ \mu_k \end{pmatrix} = \frac{1}{|\bar{A}_{k+1}|} \begin{pmatrix} g_{k+1}^T y_k g_{k+1}^T s_k - \|g_{k+1}\|^2 s_k^T y_k \\ \|g_{k+1}\|^2 g_{k+1}^T y_k - \rho_{k+1} g_{k+1}^T s_k \end{pmatrix}. \tag{2.39}$$

Moreover, due to the similarity between (2.21) and (2.39), the HS direction is still considered once the conditions (2.27) and (2.28) hold.

Hence, the search direction d_{k+1} is calculated by (1.8) and (2.39) if there only holds the condition (2.35) or by (2.29) if (2.27) and (2.28) hold, Otherwise, we exploit the negative gradient direction as our search direction.

To sum up, the generated direction possesses several forms as follows:
when the approximation conic model is selected, *i.e.* (2.1) fails,

- d_{k+1} is calculated by (2.3) and (2.8), if (2.16)–(2.20) hold,
- d_{k+1} is calculated by (1.8) and (2.21), if (2.22)–(2.26) hold,
- d_{k+1} is calculated by (2.29), if (2.27) and (2.28) hold;

when the approximation quadratic model is selected, *i.e.* (2.1) holds,

- d_{k+1} is calculated by (2.3) and (2.32), if (2.35)–(2.37) hold,
- d_{k+1} is calculated by (1.8) and (2.39), if only (2.35) holds,
- d_{k+1} is calculated by (2.29), if (2.27) and (2.28) hold;

otherwise, $d_{k+1} = -g_{k+1}$.

3. The stepsize and algorithm

In the last section, we have derived the calculations of search direction, but it is not enough for a iterative formula to solve optimization problem. This section will present the obtainment of another essential ingredient for a iterative formula, *i.e.* stepsize, which will be divided into two parts: the choice of initial stepsize and the line search procedure. And then the whole algorithm will be detailed.

3.1. strategy for the initial stepsize

It is generally acknowledged that the initial stepsize is of great significance for a optimization method, especially for conjugate gradient method. For Newton and quasi-Newton methods, the choice of the initial trial stepsize may always be unit step $\alpha^0 = 1$. However, it is different for methods that do not produce well scaled search directions, such as the steepest descent or the conjugate gradient methods. Thus, it is significant to make a reasonable initial guess of the stepsize by considering the current information about the objective function and algorithm for such methods [4,29]. Many strategies of the initial stepsize have been proposed which can be referred to [12,21,27,29].

In the strategy of [12], Dai and Kou presented a condition,

$$\frac{|\varphi_{k+1}(\alpha^0_{k+1}) - \varphi_{k+1}(0)|}{\varepsilon_1 + |\varphi_{k+1}(0)|} \le \varepsilon_2, \tag{3.1}$$

where $\varepsilon_1, \varepsilon_2$ are small positive constants, α^0_{k+1} denotes the initial trial stepsize, and $\varphi_{k+1}(\alpha) = f(x_{k+1}+\alpha d_{k+1})$. If (3.1) holds, it implies that the points $x_{k+1} + \alpha^0_{k+1} d_{k+1}$ and x_{k+1} are not far away from each other, so it is reasonable to use the minimizer of $q(\varphi_{k+1}(0), \varphi'_{k+1}(0), \varphi_{k+1}(\alpha^0_{k+1}))$ as the new initial stepsize, where $q(\varphi_{k+1}(0), \varphi'_{k+1}(0), \varphi_{k+1}(\alpha^0_{k+1}))$ is the quadratic interpolation function for $\varphi_{k+1}(0)$, $\varphi'_{k+1}(0)$ and $\varphi_{k+1}(\alpha^0_{k+1})$, and $\varphi'_{k+1}(0)$ denotes the first derivative of $\varphi_{k+1}(0)$.

In this paper, the selection of the initial stepsize has two parts depending on whether the negative gradient direction is adopted or not, and is presented *via* modification of that in [26].

When the search direction is negative gradient direction, according to the analysis of [27], it is desirable to take the initial trial stepsize by

$$\bar{\bar{\alpha}}_{k+1} = \begin{cases} \max\{\min\{s_k^T y_k/\|y_k\|^2, \lambda_{\max}\}, \lambda_{\min}\}, & \text{if } g_{k+1}^T s_k > 0, \\ \max\{\min\{\|s_k\|^2/s_k^T y_k, \lambda_{\max}\}, \lambda_{\min}\}, & \text{if } g_{k+1}^T s_k \le 0, \end{cases} \tag{3.2}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are two positive constants controlling the initial stepsize within the interval $[\lambda_{\min}, \lambda_{\max}]$ which is preferable in numerical experiments.

Andrei [2] thinks that the higher accurate the step length is, the faster convergence a conjugate gradient algorithm possesses, so it makes sense to verify if the initial trial stepsize $\bar{\bar{\alpha}}_{k+1}$ satisfies (3.1) or not. If so and $d_k \ne -g_k$, $\|g_{k+1}\|^2 \le 1$, we update the initial stepsize by

$$\tilde{\alpha}_{k+1} = \max\{\min\{\tilde{\tilde{\alpha}}_{k+1}, \lambda_{\max}\}, \lambda_{\min}\},$$

where

$$\tilde{\tilde{\alpha}}_{k+1} = \arg\min q(\varphi_{k+1}(0), \varphi'_{k+1}(0), \varphi_{k+1}(\bar{\bar{\alpha}}_{k+1})).$$

Therefore, the initial stepsize for the negative gradient direction is

$$\alpha^0_{k+1} = \begin{cases} \tilde{\alpha}_{k+1}, & \text{if (3.1) holds, } d_k \ne -g_k,\ \|g_{k+1}\|^2 < 1 \text{ and } \tilde{\tilde{\alpha}}_{k+1} > 0, \\ \bar{\bar{\alpha}}_{k+1}, & \text{otherwise.} \end{cases} \tag{3.3}$$

When it comes to the search direction that is not negative gradient direction, the similarity between the calculation of our direction and that in quasi-Newton methods implies that the unit stepsize $\alpha_{k+1}^0 = 1$ might be a reasonable initial trial stepsize. Again we figure out the minimizer of the quadratic interpolation function

$$\bar{\alpha}_{k+1} = \arg\min q(\varphi_{k+1}(0), \varphi_{k+1}^{'}(0), \varphi_{k+1}(1)).$$

If $\alpha_{k+1}^0 = 1$ satisfies (3.1) and $\bar{\alpha}_{k+1} > 0$, we update the initial stepsize by

$$\hat{\alpha}_{k+1} = \max\{\min\{\bar{\alpha}_{k+1}, \lambda_{\max}\}, \lambda_{\min}\}.$$

Therefore, the initial stepsize for the search direction except negative gradient direction is

$$\alpha_{k+1}^0 = \begin{cases} \hat{\alpha}_{k+1}, & \text{if (3.1) holds and } \bar{\alpha}_{k+1} > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{3.4}$$

3.2. the nonmonotone line search

For the variable η_k in ZH line search, Zhang and Hager [45] proved that if $\eta_{\max} = 1$, then the generated sequence $\{x_k\}$ only has the property that

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

Liu and Liu [27] presented a formula of η_k

$$\eta_k = \begin{cases} c, & \bmod (k,n) = n-1, \\ 1, & \bmod (k,n) \neq n-1, \end{cases}$$

where $0 < c < 1$ and $\bmod(k,n)$ denotes the residue for k modulo n, and resulted in a better convergence.

Referring to the above study, Li *et al.* [24] and Sun *et al.* [36] made some modification so that the improved line search is more appropriate for their algorithms.

In order to gain the decent convergence result and performance, this paper adopts the improved ZH line search used in [36]. To be specific, we set

$$C_{k+1} = \begin{cases} f_{k+1} + \min\{1,\ 0.9(C_k - f_{k+1})\}, & k < 3, \\ [\eta_k Q_k C_k + f(x_{k+1})]/Q_{k+1}, & k \geq 3, \end{cases} \tag{3.5}$$

$$Q_{k+1} = \begin{cases} 4, & k = 2, \\ \eta_k Q_k + 1, & k \geq 3, \end{cases} \tag{3.6}$$

where

$$\eta_k = \begin{cases} \eta, & \bmod (k,n) = 0, \\ 1, & \bmod (k,n) \neq 0, \end{cases} \tag{3.7}$$

where $\eta = 0.9999$.

3.3. Algorithm

In this subsection, we will detail our three-dimensional subspace method based on conic model for unconstrained optimization.

Before presenting out algorithm, we introduce a significant strategy in conjugate gradient method, restart. A restart strategy means that the search direction is recalculated by the restart direction when restart criterion is satisfied. A typical one is to reset the direction to the steepest descent direction every n iterations, since the direction after n steps is no longer conjugate for general non-quadratic function. Crowder and Wolfe [10] proved that the standard conjugate gradient method without restart reaches at most linear convergence. Yuan [41] also showed that the convergence rate of conjugate gradient method without restart is exactly linear for uniformly convex quadratics. However, the convergence rate of conjugate gradient method which is restarted with the negative gradient direction every n steps may be improved from linear to n-step quadratic [9,28]. In addition, Beale [8] proposed such restart technique that the restart direction is a combination of the negative gradient and the previous search directions which includes the second-order derivative information achieved by the search along the previous directions. Powell [31] introduced a new restart criterion implemented with Beale's method and obtained satisfactory numerical results. Considering an idea of Powell, Dai, Liao and Li [14] presented a new restart technique and designed two conjugate gradient methods based on this technique.

Algorithm 1 TSCG_Conic

Require: initial point x_0, initial stepsize α_0^0, positive constants $\epsilon,\ \epsilon_1,\ \epsilon_2,\ 0<\delta<\sigma<1, \xi_1,\ \xi_2,\ \xi_3,\ \xi_4,\ \xi_5,\ \vartheta_1,\ \vartheta_2,\ \vartheta_3 \in (0,1),\ \vartheta_4,\ \vartheta_5 \in [0,1),\ \rho_0,\ \bar{\rho}_0 \in (0,1),\ \lambda_1,\ \lambda_2,\ \lambda_{\min},\ \lambda_{\max}$

Ensure: optimal x^*

1: set MaxRestart:=4n, IterRestart:=0, IterQuad:=0, MinQuad:=3, Numnongrad:=0, $C_0 = f_0,\ d_0 = -g_0$ and $k := 0$.

2: if $\|g_0\|_\infty \le \epsilon$, stop.

3: calculate the stepsize α_k by (1.5) and (1.6) with α_k^0.

4: update $x_{k+1} = x_k + \alpha_k d_k$. If $\|g_{k+1}\|_\infty \le \epsilon$, stop; otherwise, let IterRestart:=IterRestart+1. If $|r_k - 1| \le 10^{-9}$, IterQuad:=IterQuad+1; else, IterQuad:=0.

5: calculate the search direction d_{k+1}. If Numnongrad=MaxRestart, or (IterQuad=MinQuad and IterRestart $\neq$IterQuad), go to 5.6; else if the condition (2.1) holds, or the conditions

$$(s_k^T y_k)^2 \le 10^{-6}\|s_k\|^2\|y_k\|^2 \quad \text{and} \quad \|s_k\|^2 \ge 1.5$$

hold, go to 5.1; else, go to 5.3.

5.1: if conditions (2.35)–(2.37) hold, compute d_{k+1} by (2.3) and (2.32), set Numnongrad:=Numnongrad+1; else, go to 5.2.

5.2: if condition (2.35) holds, compute d_{k+1} by (1.8) and (2.39), set Numnongrad:=Numnongrad+1; else, go to 5.5.

5.3: if conditions (2.16)–(2.20) hold, compute d_{k+1} by (2.3) and (2.8), set Numnongrad:=Numnongrad+1; else, go to 5.4.

5.4: if conditions (2.22)–(2.26) hold, compute d_{k+1} by (1.8) and (2.21), set Numnongrad:=Numnongrad+1; else, go to 5.5.

5.5: if conditions (2.27) and (2.28) hold, compute d_{k+1} by (2.29), set Numnongrad:=Numnongrad+1; else, go to 5.6.

5.6: compute $d_{k+1} = -g_{k+1}$, set Numnongrad:=0 and IterRestart:=0.

6: if $d_{k+1} = -g_{k+1}$, calculate α_{k+1}^0 by (3.3); otherwise, calculate α_{k+1}^0 by (3.4).

7: update Q_{k+1} and C_{k+1} by (3.5) and (3.6) with (3.7).

8: set $k := k+1$, go to line 3.

In this paper, we incorporate a special restart technique proposed by Dai and Kou [12]. They defined a quantity

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

where $f_{k+1} = \varphi_k(\alpha_k)$ and $f_k = \varphi_k(0)$. If r_k is close to 1, they think that the line search function φ_k is close to some quadratic function. According to their analysis, the exact approach of this quantity is that if there are continuously many iterations such that r_k is close to 1, we restart the algorithm with the negative gradient direction. In addition, if the number of the iterations since the last restart reaches the MaxRestart threshold, we also restart out algorithm.

The details of the three-dimensional subspace conjugate gradient method based on conic model are given as Algorithm 1.

In the line 5 of Algorithm 1, the conditions [24]

$$(s_k^T y_k)^2 \le 10^{-6}\|s_k\|^2\|y_k\|^2 \quad \text{and} \quad \|s_k\|^2 \ge 1.5$$

indicate that a problem seems to be ill-conditioned and the current iterative point is far away from the minimizer of the problem, which might lead to the inaccuracy of information. Thus, a quadratic model is more reliable to approximate the objective function under this circumstance.

4. Theoretical results

In this section, we will prove some important properties of the generated algorithm, including the sufficient descent property and global convergence property.

4.1. Properties of the search direction

This subsection will make deep discussion on the search directions generated by our algorithm under every situation. At first, we propose two assumptions as follows.

Assumption 4.1. *The objective function $f(x)$ is continuously differentiable and bounded from below on $\mathbb{R}^n$.*

Assumption 4.2. *The gradient function $g(x)$ is Lipschitz continuous on the level set $D = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$, which means that there exists a positive constant $L > 0$ satisfying*

$$\|g(x) - g(y)\| \le L\|x - y\|, \ \forall x, y \in D,$$

which implies that $\|y_k\| \le L\|s_k\|$.

Next, we can prove some properties of the search directions.

Lemma 4.3. *For the search direction d_{k+1} calculated by TSCG_Conic, there exists a constant $\kappa_1 > 0$ such that*

$$g_{k+1}^T d_{k+1} \le -\kappa_1 \|g_{k+1}\|^2. \tag{4.1}$$

Proof. We will discuss in several parts based on different situations and different approximation models.

Case I. if the negative direction is adopted, *i.e.* $d_{k+1} = -g_{k+1}$, then

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

Case II. if d_{k+1} is determined by (2.29), *i.e.* the HS direction, combining (2.28), then we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + \beta_k^{HS} g_{k+1}^T d_k \\ &\le -\|g_{k+1}\|^2 + \frac{|g_{k+1}^T y_k g_{k+1}^T d_k|}{d_k^T y_k} \\ &\le -\|g_{k+1}\|^2 + \vartheta_5 \|g_{k+1}\|^2 \\ &= -(1 - \vartheta_5)\|g_{k+1}\|^2. \end{aligned}$$

Case III(conic). if d_{k+1} is calculated by (1.8) and (2.21), Li *et al.* [24] have proved that

$$g_{k+1}^T d_{k+1} \le -\bar{\kappa}\|g_{k+1}\|^2,$$

where

$$\bar{\kappa} = \min\left\{\frac{\bar{\rho}_0}{(8 - 6\bar{\rho}_0 \max\{\xi_2, \xi_4\})}, \frac{1}{6\bar{c}}\right\},$$

and $\bar{c}$ is a positive constant.

Case IV(conic). when d_{k+1} is obtained by (2.3) and (2.8), we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} t_{k+1} \\ \mu_{k+1} \\ \nu_{k+1} \end{pmatrix} \\ &= -\frac{1}{D_{k+1}} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix}^T \begin{pmatrix} X & \theta_1 & \theta_2 \\ \theta_1 & \theta & \theta_3 \\ \theta_2 & \theta_3 & Y \end{pmatrix} \begin{pmatrix} \|g_{k+1}\|^2 \\ g_{k+1}^T s_k \\ g_{k+1}^T y_k \end{pmatrix} \\ &= -\frac{\|g_{k+1}\|^4}{D_{k+1}} h(x,y), \end{aligned}$$

where $x \triangleq \frac{g_{k+1}^T y_k}{\|g_{k+1}\|^2}$, $y \triangleq \frac{g_{k+1}^T s_k}{\|g_{k+1}\|^2}$. And $h(x,y)$ is a binary quadratic function of x and y which can be expressed as

$$h(x,y) = Yx^2 + 2\theta_3 xy + \theta y^2 + 2\theta_2 x + 2\theta_1 y + X.$$

It is easy to acquire the Hessian of $h(x,Y)$

$$H_h = \begin{pmatrix} 2Y & 2\theta_3 \\ 2\theta_3 & 2\theta \end{pmatrix},$$

we have $Y > 0$ because A_{k+1} is positive definite, and the determinant of H_h

$$4Y\theta - 4\theta_3^2 = 4\rho_{k+1}|A_{k+1}|,$$

is also positive, so $h(x,y)$ has a minimizer, that is

$$h(x,y)_{\min} = \frac{|A_{k+1}|}{\rho_{k+1}}.$$

Therefore, we can get

$$g_{k+1}^T d_{k+1} \le -\frac{\|g_{k+1}\|^4}{D_{k+1}} h(x,y)_{\min} \le -\frac{|A_{k+1}|}{D_{k+1}\rho_{k+1}} \|g_{k+1}\|^4. \tag{4.2}$$

Because ρ_{k+1}, D_{k+1} and $|A_{k+1}|$ are all positive, we just need to seek for the lower bound of $\frac{|A_{k+1}|}{D_{k+1}\rho_{k+1}}\|g_{k+1}\|^2$. Since D_{k+1} contains ρ_{k+1}, we first prove that ρ_{k+1} has an upper bound; that is the upper bounds of N_k, n_k and K.
For N_k, we use Cauchy inequality and can get

$$\begin{aligned} |N_k| &= |\frac{S_k}{s_k^T y_k \tau_k}|/M_k \\ &\le \Bigg[|\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \Bigg(\frac{\|g_{k+1}\|^4}{|g_{k+1}^T s_k|} + 4\frac{\|g_{k+1}\|^4 \|y_k\|^4}{|g_{k+1}^T s_k| s_k^T y_k \tau_k} + 4\frac{\|g_{k+1}\|^2 \|y_k\|^2 |\omega_k|}{s_k^T y_k \tau_k} \\ &\qquad + 2\frac{\|g_{k+1}\|^3 \|y_k\|}{s_k^T y_k} + 2\frac{\|g_{k+1}\|^3 \|y_k\| |\omega_k|}{|g_{k+1}^T s_k| \tau_k} + \frac{\|g_{k+1}\| \|s_k\| \omega_k^2}{s_k^T y_k \tau_k} \Bigg) \\ &\qquad + \frac{\|g_{k+1}\|^2 \|y_k\|^2}{s_k^T y_k} + \frac{\omega_k^2}{\tau_k} + 2\frac{\|g_{k+1}\| \|y_k\|^3 |\omega_k|}{s_k^T y_k \tau_k} \Bigg] /M_k, \end{aligned}$$

combining (2.14), (2.15) and using cauchy inequality again, we have

$$|N_k| \leq \frac{1}{M_k}\|g_{k+1}\|^2 \left[|\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \left(\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} + 4\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} + 4\frac{2+\xi_2}{\zeta_k}\frac{\|g_{k+1}\|}{\|y_k\|} \right.\right.$$
$$\left. + 2\frac{\|g_{k+1}\|\|y_k\|}{s_k^T y_k} + 2\frac{2+\xi_2}{\zeta_k}\frac{s_k^T y_k}{\|y_k\|^2}\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} + \frac{(2+\xi_2)^2}{\zeta_k}\frac{\|g_{k+1}\|\|s_k\|}{\|y_k\|^2} \right)$$
$$\left. + \frac{\|y_k\|^2}{s_k^T y_k} + \frac{(2+\xi_2)^2}{\zeta_k}\frac{s_k^T y_k}{\|y_k\|^2} + 2\frac{2+\xi_2}{\zeta_k} \right].$$

Under condition (2.18), we have $\frac{\|y_k\|\|s_k\|}{s_k^T y_k} \leq \sqrt{\frac{\xi_2}{\xi_1}}$, $\frac{\|s_k\|}{\|y_k\|} \leq \frac{1}{\xi_1}$ and

$$\frac{\|g_{k+1}\|\|y_k\|}{s_k^T y_k} = \frac{\|g_{k+1}\|\|y_k\||g_{k+1}^T s_k|}{s_k^T y_k |g_{k+1}^T s_k|} \leq \frac{\|y_k\|\|s_k\|}{s_k^T y_k}\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|},$$

in the same way,

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

Note that $\zeta_k \geq 1$, the above inequality of $|N_k|$ can be simplified to

$$|N_k| \leq \frac{1}{M_k}\|g_{k+1}\|^2 \left[|\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \left(5 + 4(2+\xi_2)\frac{\|s_k\|}{\|y_k\|} + 2\frac{\|y_k\|\|s_k\|}{s_k^T y_k} \right.\right.$$
$$\left.\left. + 2(2+\xi_2)\frac{s_k^T y_k}{\|y_k\|^2} + (2+\xi_2)^2\frac{\|s_k\|^2}{\|y_k\|^2} \right)\frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} + \left(1 + (2+\xi_2)\frac{s_k^T y_k}{\|y_k\|^2}\right)^2 \frac{\|y_k\|^2}{s_k^T y_k} \right].$$

Utilizing conditions (2.17)–(2.19) and the expression of K, we can obtain the upper bound of N_k,

$$\begin{aligned}
|N_k| &\leq \frac{\|g_{k+1}\|^2}{M_k}\left[\left(\frac{2+\xi_2}{\xi_1} + 1\right)^2 \frac{\|y_k\|^2}{s_k^T y_k} \right.\\
&\quad \left. + \left(\frac{(2+\xi_2)^2}{\xi_1^2} + 6\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 5 \right) |\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \frac{\|g_{k+1}\|^2}{|g_{k+1}^T s_k|} \right]\\
&\leq \frac{1}{\rho_0}\left(2\frac{(2+\xi_2)^2}{\xi_1^2} + 8\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 6 \right) K_1\|g_{k+1}\|^2\\
&= \left(2\frac{(2+\xi_2)^2}{\xi_1^2} + 8\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 6 \right)\frac{K}{\rho_0}.
\end{aligned}$$

Next, the upper bound of n_k is acquired by the same way,

$$\begin{aligned}
|n_k| &\leq \frac{1}{m_k}\|g_{k+1}\|^2 \left(\frac{\|y_k\|^2}{s_k^T y_k} + \frac{(2+\xi_2)^2}{\zeta_k}\frac{s_k^T y_k}{\|y_k\|^2} + 2\frac{(2+\xi_2)}{\zeta_k} \right)\\
&\leq \frac{1}{m_k}\|g_{k+1}\|^2 \left(1 + (2+\xi_2)\frac{s_k^T y_k}{\|y_k\|^2} \right)^2 \frac{\|y_k\|^2}{s_k^T y_k}\\
&\leq \frac{1}{1-\frac{1}{\zeta_k}}\frac{(\xi_1+\xi_2+2)^2}{\xi_1^2}K_1\|g_{k+1}\|^2\\
&\leq 6\frac{(\xi_1+\xi_2+2)^2}{\xi_1^2}K.
\end{aligned}$$

For K_1, it is easy to know that

$$K_1 \leq \max\{\xi_2, \xi_3\}.$$

Through the above discussion, now we can give the upper bound of ρ_{k+1},

$$\begin{aligned}
\rho_{k+1} &= \zeta_k \max\{K, N_k, n_k\} \\
&\leq 2K_1 \|g_{k+1}\|^2 \times \\
&\qquad \max\{(2\frac{(2+\xi_2)^2}{\xi_1^2} + 8\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 6)/\rho_0,\ 6\frac{(\xi_1+\xi_2+2)^2}{\xi_1^2}\} \\
&\leq 2\max\{\xi_2, \xi_3\}\|g_{k+1}\|^2 \times \\
&\qquad \max\{(2\frac{(2+\xi_2)^2}{\xi_1^2} + 8\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 6)/\rho_0,\ 6\frac{(\xi_1+\xi_2+2)^2}{\xi_1^2}\},
\end{aligned}$$

for convenience, we define

$$L_0 \triangleq 2\max\{(2\frac{(2+\xi_2)^2}{\xi_1^2} + 8\frac{(2+\xi_2)}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 6)/\rho_0,\ 6\frac{(\xi_1+\xi_2+2)^2}{\xi_1^2}\}\max\{\xi_2, \xi_3\},$$

namely, $\rho_{k+1} \leq L_0\|g_{k+1}\|^2$.

Since we have found the upper bound of ρ_{k+1}, then it turns to that of D_{k+1}. According to (2.12), D_k can be expressed as

$$D_{k+1} = s_k^T y_k \tau_k \rho_{k+1} M_k - S_k = s_k^T y_k \tau_k (M_k \rho_{k+1} - M_k N_k). \tag{4.3}$$

Using the formula of M_k and the upper bounds of ρ_{k+1}, we have

$$\begin{aligned}
|D_{k+1}| \leq \|g_{k+1}\|^2 s_k^T y_k \tau_k \Bigg[L_0 + L_0 \frac{\|y_k\|^4}{s_k^T y_k \tau_k} + L_0 |\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \Bigg(2\frac{\|g_{k+1}\|\|y_k\|^3}{s_k^T y_k \tau_k} \\
+ \frac{\|g_{k+1}\|\|s_k\|}{s_k^T y_k} + \frac{\|g_{k+1}\|^2\|y_k\|^2}{|g_{k+1}s_k|\tau_k} \Bigg) + \frac{M_k|N_k|}{\|g_{k+1}\|^2} \Bigg].
\end{aligned}$$

Because

$$\frac{\|g_{k+1}\|\|y_k\|^3}{s_k^T y_k \tau_k} = \frac{\|g_{k+1}\|\|y_k\|^3|g_{k+1}s_k|}{s_k^T y_k \tau_k |g_{k+1}s_k|} \leq \frac{\|s_k\|}{\|y_k\|}\frac{\|g_{k+1}\|^2}{|g_{k+1}s_k|},$$

$$\frac{\|g_{k+1}\|\|s_k\|}{s_k^T y_k} = \frac{\|g_{k+1}\|\|s_k\||g_{k+1}s_k|}{s_k^T y_k |g_{k+1}s_k|} \leq \frac{\|s_k\|^2}{s_k^T y_k}\frac{\|g_{k+1}\|^2}{|g_{k+1}s_k|},$$

and with the upper bound of N_k, the above inequality of D_{k+1} can be simplified to

$$\begin{aligned}
D_{k+1} \leq \|g_{k+1}\|^2 s_k^T y_k \tau_k \Bigg[2L_0 + L_0 \left(2\frac{\|s_k\|}{\|y_k\|} + \frac{\|s_k\|^2}{s_k^T y_k} + \frac{s_k^T y_k}{\|y_k\|^2} \right) |\frac{1-\gamma_{k+1}}{\gamma_{k+1}}| \frac{\|g_{k+1}\|^2}{|g_{k+1}s_k|} \\
+ (4\frac{\xi_2}{\xi_1} + 6\sqrt{\frac{\xi_2}{\xi_1}} + 16)K_1 \Bigg].
\end{aligned}$$

Finally, using (2.18) and (2.19), we can get

$$\begin{aligned} D_{k+1} &\le \|g_{k+1}\|^2 s_k^T y_k \tau_k \left[2L_0 + (4\frac{L_0}{\xi_1} + 4\frac{\xi_2}{\xi_1} + 6\sqrt{\frac{\xi_2}{\xi_1}} + 16)K_1\right] \\ &\le \|g_{k+1}\|^2 s_k^T y_k \tau_k \left[2L_0 + (4\frac{L_0}{\xi_1} + 4\frac{\xi_2}{\xi_1} + 6\sqrt{\frac{\xi_2}{\xi_1}} + 16)\max\{\xi_2, \xi_3\}\right] \\ &\le L_1 \|g_{k+1}\|^2 s_k^T y_k \tau_k, \end{aligned} \tag{4.4}$$

where

$$L_1 \triangleq \left[2L_0 + (4\frac{L_0}{\xi_1} + 4\frac{\xi_2}{\xi_1} + 6\sqrt{\frac{\xi_2}{\xi_1}} + 16)\max\{\xi_2, \xi_3\}\right].$$

With (2.20), (4.2) and (4.4), the sufficient descent property of d_{k+1} under this case can be established by

$$g_{k+1}^T d_{k+1} \le -\frac{\xi_5}{L_1}\|g_{k+1}\|^2.$$

Case V(quadratic). when d_{k+1} is calculated by (2.3) and (2.32), or by (1.8) and (2.39), we can prove that

$$g_{k+1}^T d_{k+1} \le -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}. \tag{4.5}$$

Firstly, if d_{k+1} is generated by (1.8) and (2.39), the proof can be referred to [13]. Secondly, if d_{k+1} is generated by (2.3) and (2.32), we can take the same approach as last case to obtain

$$g_{k+1}^T d_{k+1} = -\frac{\|g_{k+1}\|^4}{|A_{k+1}|} h(x, y).$$

Since we have figured out that

$$h(x, y)_{\min} = \frac{|A_{k+1}|}{\rho_{k+1}},$$

it is obvious that

$$g_{k+1}^T d_{k+1} \le -\frac{\|g_{k+1}\|^4}{\rho_{k+1}}.$$

With the property (4.5), we are able to give the upper bound of $g_{k+1} d_{k+1}$ by finding the upper bound of ρ_{k+1}. Based on the expression of ζ_k and (2.33), it is easy to get

$$\rho_{k+1} = \zeta_k \frac{\|y_k\|^2}{s_k^T y_k}\|g_{k+1}\|^2 \le 2\vartheta_2 \|g_{k+1}\|^2. \tag{4.6}$$

Combining it with (4.5), we finally have

$$g_{k+1}^T d_{k+1} \le -\frac{1}{2\vartheta_2}\|g_{k+1}\|^2.$$

After the above discussion, we can prove that there exists a constant κ_1 such that

$$g_{k+1}^T d_{k+1} \le -\kappa_1 \|g_{k+1}\|^2,$$

where

$$\kappa_1 = \min\{\bar{\kappa}, \frac{1}{2}, 1-\vartheta_5, \frac{\xi_5}{L_1}, \frac{1}{2\vartheta_2}\}.$$

The proof is complete.

□

Lemma 4.4. *Assume that f satisfies Assumption 4.2. If the search direction d_{k+1} is calculated by TSCG_Conic, then there exists a constant $\kappa_2 > 0$ such that*

$$\|d_{k+1}\| \leq \kappa_2 \|g_{k+1}\|. \tag{4.7}$$

Proof. Similar to Lemma 4.3, the proof is divided into several cases.

Case I. if d_{k+1} is calculated by negative gradient, then there certainly holds

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

Case II. if the search direction is HS direction, according to Assumption 4.2 and the condition (2.27), we have

$$\begin{aligned}\|d_{k+1}\| &= \|-g_{k+1} + \beta_k^{HS} d_k\| \\ &\leq \|g_{k+1}\| + \frac{\|g_{k+1}\|\|y_k\|\|d_k\|}{d_k^T y_k} \\ &\leq \left(1 + \frac{L}{\vartheta_1}\right)\|g_{k+1}\|.\end{aligned}$$

Case III(conic). if d_{k+1} is calculated by (1.8) and (2.21), Li *et al.* [24] have proved that

$$\|d_{k+1}\| \leq \left(\frac{10\xi_2 + 5\xi_1 + 5m}{\bar{\rho}_0 \xi_1^2}\right)\|g_{k+1}\|,$$

where $m = 2n_0\bar{\xi}/\bar{\rho}_0$ with

$$n_0 = \max\left\{4 - 3\bar{\rho}_0, 1 + \frac{2\xi_2}{\xi_1}\right\}, \quad \bar{\xi} = \max\{\xi_2, \xi_3, \xi_4\}.$$

Case IV(conic). if d_{k+1} is formed by (2.3) and (2.8), we have

$$\begin{aligned}\|d_{k+1}\| &= \|t_{k+1}g_{k+1} + \mu_{k+1}s_k + \nu_{k+1}y_k\| \\ &\leq \frac{1}{D_{k+1}}(|q_1|\|g_{k+1}\| + |q_2|\|s_k\| + |q_3|\|y_k\|),\end{aligned} \tag{4.8}$$

so in order to prove Lemma 4.4 under this case, we first need to obtain the lower bound of D_{k+1}. Combining (2.13), (2.17), (4.3), and the value range of ζ_k, we can derive

$$\begin{aligned}D_{k+1} &= s_k^T y_k \tau_k M_k(\zeta_k \max\{K, N_k, n_k\} - N_k) \\ &\geq s_k^T y_k \tau_k \rho_0 \left(\frac{6}{5}\max\{K, N_k, n_k\} - N_k\right) \\ &\geq s_k^T y_k \tau_k \rho_0 \frac{K}{5}.\end{aligned}$$

Then using the above inequality, we can transform (4.8) into

$$\begin{aligned}
\|d_{k+1}\| &\leq \frac{5}{\rho_0 K s_k^T y_k \tau_k}(|q_1|\|g_{k+1}\| + |q_2|\|s_k\| + |q_3|\|y_k\|) \\
&\leq \frac{5}{\rho_0 K}\|g_{k+1}\| \Bigg[\|g_{k+1}\|^2 + 4\frac{\|g_{k+1}\|^2\|y_k\|^4}{s_k^T y_k \tau_k} + 2\frac{\|g_{k+1}\|^2\|s_k\|\|y_k\|}{s_k^T y_k} \\
&\qquad + 4\frac{\|g_{k+1}\|\|s_k\|\|y_k\|^2|\omega_k|}{s_k^T y_k \tau_k} + 2\frac{\|g_{k+1}\|\|y_k\||\omega_k|}{\tau_k} + \frac{\rho_{k+1}\|s_k\|^2}{s_k^T y_k} \\
&\qquad + \frac{\|s_k\|^2\omega_k^2}{s_k^T y_k \tau_k} + 2\frac{\rho_{k+1}\|s_k\|\|y_k\|^3}{s_k^T y_k \tau_k} + \frac{\rho_{k+1}\|y_k\|^2}{\tau_k}\Bigg] \\
&\leq \frac{5\|g_{k+1}\|}{\rho_0}\frac{\|g_{k+1}\|^2}{K}\left(\frac{(2+\xi_2)^2}{\xi_1^2} + 6\frac{(2+\xi_2)}{\xi_1} + 4\frac{M_0}{\xi_1} + 2\sqrt{\frac{\xi_2}{\xi_1}} + 5\right) \\
&\leq \frac{\|g_{k+1}\|}{\rho_0 K_1}\frac{25\xi_1^2 + 5\xi_2^2 + 30\xi_1\xi_2 + 10\xi_1\sqrt{\xi_1\xi_2} + (60+20M_0)\xi_1 + 20\xi_2 + 20}{\xi_1^2}.
\end{aligned}$$

Since (2.18) implies that $K_1 \geq \xi_1$, we finally obtain the upper bound of d_{k+1}, that is

$$\|d_{k+1}\| \leq \frac{25\xi_1^2 + 5\xi_2^2 + 30\xi_1\xi_2 + 10\xi_1\sqrt{\xi_1\xi_2} + (60+20M_0)\xi_1 + 20\xi_2 + 20}{\rho_0\xi_1^3}\|g_{k+1}\|.$$

For convenience, we define

$$L_2 \triangleq \left(\frac{25\xi_1^2 + 5\xi_2^2 + 30\xi_1\xi_2 + 10\xi_1\sqrt{\xi_1\xi_2} + (60+20M_0)\xi_1 + 20\xi_2 + 20}{\rho_0\xi_1^3}\right),$$

namely, $d_{k+1} \leq L_2\|g_{k+1}\|$.

Case V(quadratic). if d_{k+1} is formed by (1.8) and (2.39), it is similar to Lemma 4 in [23]. According to their proof, we can get

$$\|d_{k+1}\| \leq \frac{20}{\vartheta_1}\|g_{k+1}\|.$$

Case VI(quadratic). if d_{k+1} is calculated by (2.3) and (2.32), we have

$$\|d_{k+1}\| = \frac{1}{|A_{k+1}|}(|q_1|\|g_{k+1}\| + |q_2|\|s_k\| + |q_3|\|y_k\|).$$

Therefore, it is necessary to seek for the lower bound of $|A_{k+1}|$. By combining (2.34) with (2.35) and (2.36), we can easily acquire

$$|A_{k+1}| \geq \frac{6}{25}\vartheta_1\vartheta_3\|g_{k+1}\|^2\|y_k\|^4.$$

According to the above inequality and Cauchy inequality, it follows that,

$$\begin{aligned}
\|d_{k+1}\| \leq \frac{25}{6\vartheta_1\vartheta_3\|g_{k+1}\|^2\|y_k\|^4}&\Bigg[\rho_{k+1}\|g_{k+1}\|\|y_k\|^4\left(2\frac{\|s_k\|^2}{s_k^T y_k} + 3\frac{\|s_k\|}{\|y_k\|}\right) \\
&+ \|g_{k+1}\|^3\|y_k\|^4\left(10 + 12\frac{\|s_k\|\|y_k\|}{s_k^T y_k} + 4\frac{\|s_k\|^2\|y_k\|^2}{(s_k^T y_k)^2}\right)\Bigg].
\end{aligned}$$

Based on (4.6) and (2.35), it implies that

$$\begin{aligned}\|d_{k+1}\| &\leq \frac{25\|g_{k+1}\|}{6\vartheta_1\vartheta_3}\left(10+12\sqrt{\frac{\vartheta_2}{\vartheta_1}}+4\frac{\vartheta_2}{\vartheta_1}+4\frac{\vartheta_2}{\vartheta_1}+6\frac{\vartheta_2}{\vartheta_1}\right)\\ &\leq \frac{125\vartheta_1+150\sqrt{\vartheta_1\vartheta_2}+175\vartheta_2}{3\vartheta_1^2\vartheta_3}\|g_{k+1}\|,\end{aligned}$$

likewise, we set

$$L_3 \triangleq \frac{125\vartheta_1+150\sqrt{\vartheta_1\vartheta_2}+175\vartheta_2}{3\vartheta_1^2\vartheta_3}.$$

According to the above analysis, the proof of Lemma 4.4 is completed by setting

$$\kappa_2 = \max\left\{\left(\frac{10\xi_2+5\xi_1+5m}{\bar{\rho}_0\xi_1^2}\right), 1, 1+\frac{L}{\vartheta_1}, L_2, \frac{20}{\vartheta_1}, L_3\right\}.$$

□

4.2. Convergence analysis

In this subsection, we will give the global convergence of the presented algorithm for general functions.

Lemma 4.5. *Suppose α_k is generated by line search (1.5) and (1.6), and $f(x)$ satisfies Assumption 4.2, then*

$$\alpha_k \geq \frac{(1-\sigma)|g_k^T d_k|}{L\|d_k\|^2}. \tag{4.9}$$

Proof. According to the line search (1.5), we can easily obtain

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

because $\sigma - 1 < 0$ and $g_k^T d_k < 0$, it is obvious that

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

□

Theorem 4.6. *Suppose the objective function $f(x)$ satisfies Assumption 4.1 and Assumption 4.2, and sequence $\{x_k\}$ is generated by TSCG_Conic, then we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{4.10}$$

Proof. According to (4.9), (4.1) and (4.7), it follows

$$\delta\alpha_k g_k^T d_k \leq -\frac{(1-\sigma)\delta}{L}\frac{(g_k^T d_k)^2}{\|d_k\|^2} \leq -\frac{(1-\sigma)\delta\kappa_1^2}{L\kappa_2}\|g_k\|^2 = -T\|g_k\|^2,$$

where $T \triangleq \frac{(1-\sigma)\delta\kappa_1^2}{L\kappa_2}$, then combining (1.6), we have

$$f_{k+1} \leq C_k + \delta\alpha_k g_k^T d_k \leq C_k - T\|g_k\|^2. \tag{4.11}$$

Notice (3.6) and (3.7), although Q_0, Q_1, Q_2 and η_0, η_1, η_2 are omitted in comparison to the standard ZH line search, we could assume that $Q_0 = 1$, $Q_1 = 2$, $Q_2 = 3$ and $\eta_0 = \eta_1 = \eta_2 = 1$. For large scale problem, its dimension n is tremendous so that $\mod(k, n) \neq 0$ always holds for small k, hence the assumption of η_0, η_1 and η_2 is reasonable according to the update formula of η_k. Moreover, all these quantities are in accord with the update formula of Q_{k+1} in (1.7), hence such an assumption is desirable in order to simplify the computation of Q_{k+1}. Therefore, from (3.6) it is easy to acquire that

$$Q_{k+1} = 1 + \sum_{j=0}^{k} \prod_{i=0}^{j} \eta_{k-i}, \quad k = 0, 1, 2 \ldots$$

Combining (3.7), we can derive the general formula of Q_{k+1},

$$Q_{k+1} = \begin{cases} 1 + \eta^{k/n} + n \sum_{i=1}^{k/n} \eta^i, & \mathrm{mod}(k, n) = 0, \\ 1 + \mathrm{mod}(k, n) + \eta^{\lfloor k/n \rfloor} + n \sum_{i=1}^{\lfloor k/n \rfloor} \eta^i, & \mathrm{mod}(k, n) \neq 0, \end{cases}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Thus, it is easy to get the upper bound of Q_{k+1},

$$\begin{aligned} Q_{k+1} &\leq 1 + \mathrm{mod}(k, n) + 1 + \eta^{\lfloor k/n \rfloor + 1} + n \sum_{i=1}^{\lfloor k/n \rfloor + 1} \eta^i \\ &\leq 1 + n + (n+1) \sum_{i=1}^{\lfloor k/n \rfloor + 1} \eta^i \\ &\leq 1 + n + (n+1) \sum_{i=1}^{k+1} \eta^i \\ &\leq (1+n) \sum_{i=0}^{k+1} \eta^i \\ &\leq \frac{1+n}{1-\eta} \\ &= H, \end{aligned} \tag{4.12}$$

where $H = \frac{1+n}{1-\eta}$.

When $k \geq 3$, Combining (1.7), (4.11) and (4.12), we get

$$C_{k+1} = C_k + \frac{f_{k+1} - C_k}{Q_{k+1}} \leq C_k - \frac{T}{H} \|g_k\|^2,$$

which means

$$\frac{T}{H} \|g_k\|^2 \leq C_k - C_{k+1}. \tag{4.13}$$

When $k < 3$, (1.6) and (3.5) implies that

$$C_{k+1} \leq f_{k+1} + 0.9(C_k - f_{k+1}) = C_k + 0.1(f_{k+1} - C_k) < C_k.$$

Hence C_k is monotonically decreasing. According to the Lemma 1.1 in [45], we have $f_{k+1} \leq C_{k+1}$ for each $k \geq 3$, and (3.5) indicates that $f_{k+1} \leq C_{k+1}$ for each $k < 3$. Thus, with the assumption that f is bounded below, C_k is certainly bounded from below.

Summing up the above analysis and (4.13), we can finally obtain that

$$\sum_{k=0}^{\infty} \frac{T}{H} \|g_k\|^2 < \infty,$$

which implies

$$\lim_{k\to\infty} \|g_k\|^2 = 0.$$

The proof is completed. □

Under the assumptions that objective function is continuously differentiable, bounded from below and gradient function is Lipschitz continuous, we have established the sufficient descent property of our directions. Moreover, our algorithm possesses the global convergence that

$$\lim_{k\to\infty} \|g_k\|^2 = 0.$$

Since we will compare our algorithm with SMCG_Conic [24], CONIC_CG3 [36] and SMCG_BB [26]in numerical experiments, we briefly introduce the convergence properties of these algorithms. Under the same assumptions, the directions of SMCG_Conic, CONIC_CG3 and SMCG_BB also satisfy the sufficient descent property, and SMCG_Conic shares the same global convergence with TSCG_Conic that $\lim_{k\to\infty} \|g_k\|^2 = 0$ while CONIC_CG3 and SMCG_BB only have the property that

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

But SMCG_BB can achieve the property that $\lim_{k\to\infty} \|g_k\|^2 = 0$ if the objective function is convex.

5. Numerical results

In this section, the results of the numerical experiments are showed below. The unconstrained test functions were taken from [1] with the given initial points. To prove the efficiency of the proposed TSCG_Conic algorithm, we compare its numerical performance with SMCG_Conic, CONIC_CG3 and SMCG_BB. And the performance profile proposed by Dolan and Moré [18] is used to evaluate the performance of these methods. The dimension of the test functions is 10,000. All the programs were written in C code.

For each problem p in the test set $\mathcal{P}$ and each solver s in the solver set $\mathcal{S}$, [18] defines the performance ratio

$$r(p,s) = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}},$$

where

$$t_{p,s} = \text{CPU time required to solve problem } p \text{ by solver } s.$$

It is evident that $r(p,s) \geq 1$, and the equality holds if and only if solver s solves problem p with the least computing time. Based on $r(p,s)$, Dolan and Moré defines

$$P_{p:r(p,s)\leq\tau} = \frac{1}{n_p} \text{size}\{p \in \mathcal{P} : r(p,s) \leq \tau\},$$

which gives an overall assessment of the performance of the solver s. The function $P_{p:r(p,s)\leq\tau}$ depicts the probability for solver s of which the performance ratio $r(p,s)$ is within a factor τ, and is the cumulative distribution function for the performance ratio. In addition to CPU time, this function can be used with other measures, including the number of iterations, function evaluations and gradient evaluations.

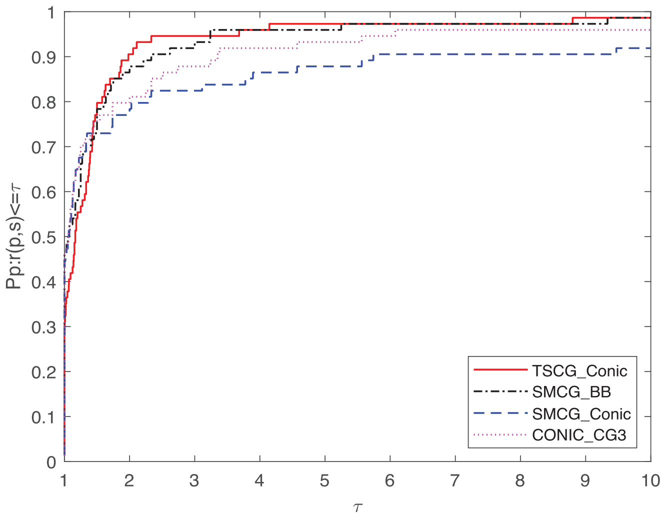


FIGURE 1. Performance profile based on the number of iterations.

SMCG_Conic is a two-dimensional subspace minimization conjugate gradient algorithm based on conic model, and is a pioneer one that combines subspace technique with conic model to seek for the search direction. The numerical experiments in [24] show that the performance of SMCG_Conic is very efficient. Since the biggest difference between TSCG_Conic and SMCG_Conic is the dimension of the used subspace, the comparison between TSCG_Conic and SMCG_Conic can not only reflect the high efficiency of our algorithm, but also reveal the influence to the numerical result due to the change of dimension for the adopted subspace in the subspace minimization conjugate gradient algorithm.

Combining the subspace $\Omega_{k+1} = Span\{g_{k+1}, s_k, s_{k-1}\}$, [36] develops a three-dimensional subspace minimization conjugate gradient algorithm based on conic model CONIC_CG3, and has shown its high efficiency. Our algorithm TSCG_Conic is also based on three-dimensional subspace and the difference is construction of subspace, hence we compare it with CONIC_CG3.

As for SMCG_BB, it is an efficient subspace minimization conjugate gradient method, and successfully apply the idea of BB method and BBCG method which are employed in our algorithm as well. Besides, our scheme of the choice for initial stepsize is a modification of that in SMCG_BB. Thus, it is also meaningful to compare the numerical performance of TSCG_Conic and SMCG_BB.

For the initial stepsize of the first iteration, we adopt the adaptive strategy used in [23]. The other parameters of TSCG_Conic are selected as follows.

$$\begin{aligned}
&\epsilon = 10^{-6},\ \epsilon_1 = 10^{-3},\ \epsilon_2 = 10^{-4},\ \delta = 0.001,\ \sigma = 0.9999,\\
&\lambda_{\min} = 10^{-30},\ \lambda_{\max} = 10^{30},\ \lambda_1 = 7 \times 10^{-8},\ \lambda_2 = 0.05,\ \rho_0 = 0.3,\ \bar{\rho}_0 = 0.9,\\
&\xi_1 = 0.2 \times 10^2,\ \xi_2 = 8 \times 10^4,\ \xi_3 = 4 \times 10^8,\ \xi_4 = 7 \times 10^7,\ \xi_5 = 0.1,\\
&\vartheta_1 = 5 \times 10^{-7},\ \vartheta_2 = 3 \times 10^3,\ \vartheta_3 = 0.9,\ \vartheta_4 = 7 \times 10^{-3},\ \vartheta_5 = 10^{-5}.
\end{aligned}$$

SMCG_Conic, CONIC_CG3 and SMCG_BB use the original parameters in their papers respectively. In addition to stopping when the stopping criterion $\|g_k\|_\infty \leq \varepsilon$ holds, the algorithm also stops when the number of iterations exceeds 200,000.

Figure 1 depicts the performance based on the number of iterations for the three methods. It shows that TSCG_Conic performs better than three other algorithms, although it is a little inferior to them when $\tau < 1.4$.

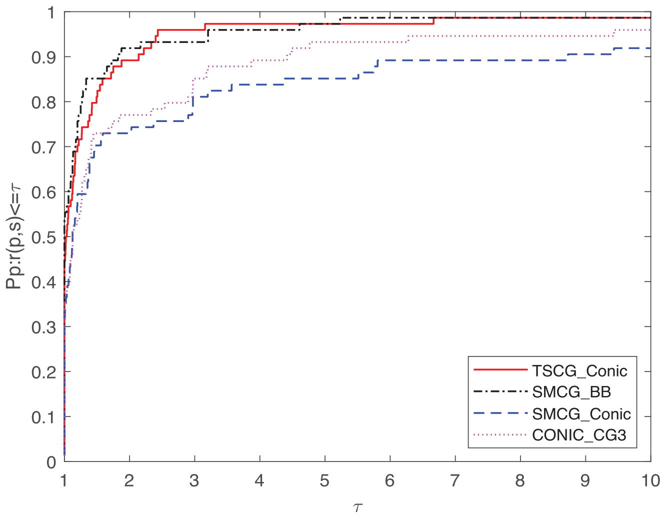


FIGURE 2. Performance profile based on the number of function evaluations.

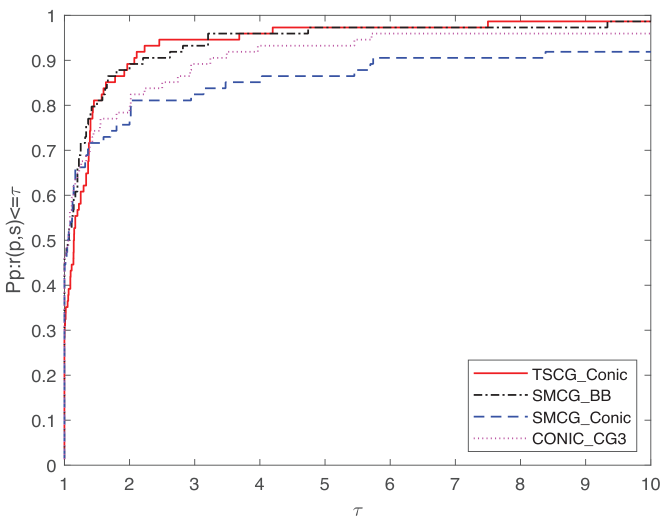


FIGURE 3. Performance profile based on the number of gradient evaluations.

In Figure 2, we observe that TSCG_Conic and SMCG_BB outperforms SMCG_Conic and CONIC_CG3 on the number of function evaluations. Besides, TSCG_Conic and SMCG_BB can solve about 50% of test problems with the least number of function evaluations, while SMCG_Conic and CONIC_CG3 solve about 35%.

Similar to Figures 1 and 3 illustrates that TSCG_Conic lags behind in comparison with three other algorithms when $\tau < 1.4$, but is competitive to SMCG_BB and superior to SMCG_Conic and CONIC_CG3 when $\tau > 1.4$.

As regards the CPU time, we can see from Figure 4 that TSCG_Conic and SMCG_BB has an appreciable improvement on SMCG_Conic and CONIC_CG3 when $\tau > 1.2$, which shows the high efficiency of TSCG_Conic.

For the 80 test problems, the numerical results show that while TSCG_Conic may perform a little worse than other algorithms for the case that τ is very small, it overall has significant improvements over SMCG_Conic and CONIC_CG3, and is competitive to SMCG_BB. Besides, TSCG_Conic can solve more problems than

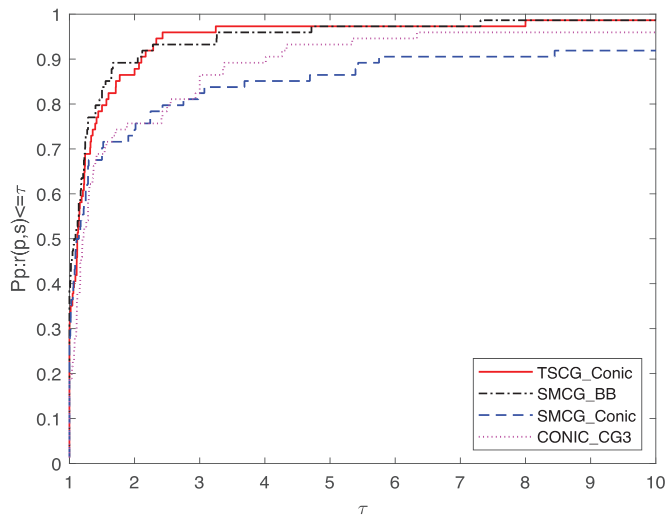
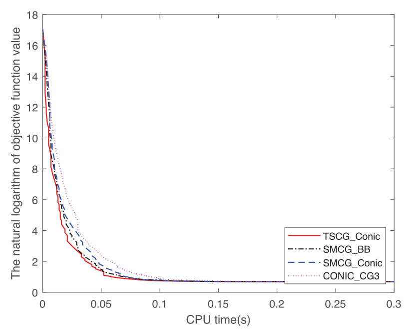
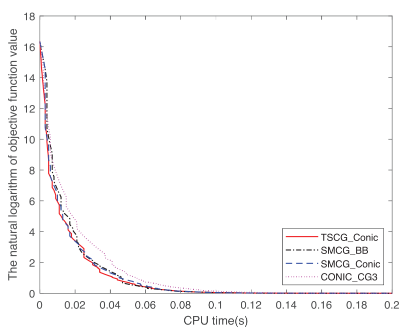


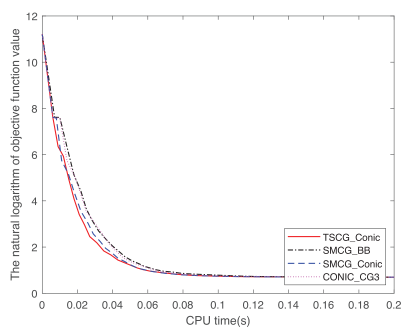
FIGURE 4. Performance profile based on the CPU time.



(a)



(b)



(c)

FIGURE 5. The natural logarithm of objective function value *vs.* CPU time.

SMCG_Conic, which might show the advantage of the three-dimensional subspace minimization method compared with the two-dimensional one. In a word, the TSCG_Conic is an efficient algorithm for solving unconstrained optimization problem.

To present the performance of these algorithms in numerical experiments more intuitively, we select three test functions and depict the figures about the objective function value *vs.* CPU time in Figure 5. For better visualization, we will use the natural logarithm of the objective function value as the y-axis.

6. Conclusion

(i) This paper has proposed a new three-dimensional subspace minimization conjugate gradient method based on conic model, the sufficient descent property of the search direction and the global convergence of this method are obtained under some suitable assumptions.
(ii) The selection of approximation model is alternative depending on whether certain criterions are satisfied. Besides, the estimates of some quantities containing B_{k+1} are various. The strategies of initial stepsize and nonmonotone line search are exploited which are beneficial to the convergence and efficiency.
(iii) From the numerical results and theoretical analysis, TSCG_Conic is competitive and promising.

Acknowledgements. Funded by Natural Science Foundation of Guangxi Province (CN) no. 2020GXNSFAA159014.

References

[1] N. Andrei, An unconstrained optimization test functions collection. *Adv. Model. Optim.* **10** (2008) 147–161.
[2] N. Andrei, Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization. *Bull. Malays. Math. Sci. Soc.* **34** (2011) 319–330.
[3] N. Andrei, An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. *Numer. Algorithms* **65** (2014) 859–874.
[4] N. Andrei, Nonlinear conjugate gradient methods for unconstrained optimization, Springer Optimization and Its Applications 158. Springer, Cham (2020).
[5] K.A. Ariyawansa, Deriving collinear scaling algorithms as extensions of quasi-Newton methods and the local convergence of DFP- and BFGS-related collinear scaling algorithms. *Math. Program.* **49** (1990) 23–48.
[6] J. Bai, W. W. Hager and H. Zhang, An inexact accelerated stochastic ADMM for separable convex optimization. *Comput. Optim. Appl.* **81** (2022) 479–518.
[7] J. Barzilai and J.M. Borwein, Two-point step size gradient methods. *IMA J. Numer. Anal.* **8** (1988) 141–148.
[8] E.M.L. Beale, A derivation of conjugate gradients. In: *Numerical Methods for Nonlinear Optimization.* Academic Press, London (1972) 39–43.
[9] A.I. Cohen, Rate of convergence of several conjugate gradient algorithms. *SIAM J. Numer. Anal.* **9** (1972) 248–259.
[10] H. Crowder and P. Wolfe, Linear convergence of the conjugate gradient method. *IBM J. Res. Dev.* **16** (1972) 431–433.
[11] Y.H. Dai and Y.X. Yuan, A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10** (1999) 177–182.
[12] Y.H. Dai and C.X. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **23** (2013) 296–320.
[13] Y.H. Dai and C.X. Kou, A Barzilai-Borwein conjugate gradient method. *Sci. China Math.* **59** (2016) 1511–1524.
[14] Y.H. Dai, L.Z. Liao and D. Li, On restart procedures for the conjugate gradient method. *Numer. Algorithms* **35** (2004) 249–260.
[15] Y.H. Dai, J.Y. Yuan and Y.X. Yuan, Modified two-point stepsize gradient methods for unconstrained optimization problems. *Comput. Optim. Appl.* **22** (2002) 103–109.
[16] W.C. Davidon, Conic approximations and collinear scalings for optimizers. *SIAM J. Numer. Anal.* **17** (1980) 268–281.
[17] S. Di and W. Sun, A trust region method for conic model to solve unconstraind optimizaions. *Optim. Methods Softw.* **6** (1996) 237–263.
[18] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **91** (2002) 201–213.
[19] R. Fletcher, Practical methods of optimization. Vol. 1: Unconstrained optimization. Wiley-Interscience, New York (1980).
[20] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients. *Comput. J.* **7** (1964) 149–154.
[21] W.W. Hager and H. Zhang, Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Softw.* **32** (2006) 113–137.
[22] M.R. Hestenes and E.L. Stiefel, Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49** (1952) 409–436.
[23] M. Li, H. Liu and Z. Liu, A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization. *Numer. Algorithms* **79** (2018) 195–219.

[24] Y. Li, Z. Liu and H. Liu, A subspace minimization conjugate gradient method based on conic model for unconstrained optimization. *Comput. Appl. Math.* **38** (2019).

[25] Y.L. Liu and C.S. Storey, Efficient generalized conjugate gradient algorithms. I: Theory. *J. Optim. Theory Appl.* **69** (1991) 129–137.

[26] Z. Liu and H. Liu, An efficient gradient method with approximate optimal stepsize for large-scale unconstrained optimization. *Numer. Algorithms* **78** (2018) 21–39.

[27] H. Liu and Z. Liu, An efficient Barzilai-Borwein conjugate gradient method for unconstrained optimization. *J. Optim. Theory Appl.* **180** (2019) 879–906.

[28] G.P. McCormick and K. Ritter, Alternative proofs of the convergence properties of the conjugate-gradient method. *J. Optim. Theory Appl.* **13** (1975) 497–518.

[29] J. Nocedal and S.J. Wright, Numerical Optimization. Springer, New York (2006).

[30] B.T. Polyak, The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **9** (1969) 94–112.

[31] M.J.D. Powell, Restart procedures for the conjugate gradient method. *Math. Program.* **12** (1977) 241–254.

[32] S. Sheng, Interpolation by conic model for unconstrained optimization. *Computing* **54** (1995) 83–98.

[33] D.C. Sorensen, The Q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization. *SIAM J. Numer. Anal.* **17** (1980) 84–114.

[34] W. Sun, On nonquadratic model optimization methods. *Asia. Pac. J. Oper. Res.* **13** (1996) 43–63.

[35] W. Sun and Y. Yuan, Optimization Theory and Methods: Nonlinear Programming. Springer, New York (2006).

[36] W. Sun, Y. Li, T. Wang and H. Liu, A new subspace minimization conjugate gradient method based on conic model for large-scale unconstrained optimization. *Comput. Appl. Math.* **41** (2022).

[37] P. Wolfe, Convergence conditions for ascent methods. *SIAM Rev.* **11** (1969) 226–235.

[38] P. Wolfe, Convergence conditions for ascent methods. II: Some corrections. *SIAM Rev.* **13** (1971) 185–188.

[39] Y. Yang, Y. Chen and Y. Lu, A subspace conjugate gradient algorithm for large-scale unconstrained optimization. *Numer. Algorithms* **76** (2017) 813–828.

[40] Y.X. Yuan, A modified BFGS algorithm for unconstrained optimization. *IMA J. Numer. Anal.* **11** (1991) 325–332.

[41] Y.X. Yuan, Analysis on the conjugate gradient method. *Optim. Methods Softw.* **2** (1993) 19–29.

[42] Y.X. Yuan, Subspace techniques for nonlinear optimization. In: *Some Topics in Industrial and Applied Mathematics*. Higher Education Press, Beijing (2007) 206–218.

[43] Y.X. Yuan, A review on subspace methods for nonlinear optimization. In: *Proc. of the International Congress of Mathematics*. Seoul, Korea, 13-21 August (2014) 807–827.

[44] Y.X. Yuan and J. Stoer, A subspace study on conjugate gradient algorithms. *Z. Angew. Math. Mech.* **75** (1995) 69–77.

[45] H. Zhang and W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.* **14** (2004) 1043–1056.