A New First-order Framework for Orthogonal Constrained Optimization Problems

Bin Gao\(^*\) Xin Liu\(^†\) Xiaojun Chen\(^‡\) Ya-xiang Yuan\(^§\)

September 30, 2016

Abstract. In this paper, we consider a class of orthogonal constrained optimization problems, the feasible region of which is called the Stiefel manifold. Our new proposed framework combines a function value reduction step with a symmetrization step. Different from the existing approaches, the function value reduction is conducted in the Euclidean space instead of the Stiefel manifold or its tangent space. We construct two types of algorithms based on this new framework. The first type is gradient reduction based algorithms which consists of gradient reflection (GR) and gradient projection (GP) two implementations. The other one adopts a column-wise block coordinate descent (CBCD) scheme with a novel idea for solving the corresponding CBCD subproblem inexactly. Theoretically, we can prove that both GR/GP with a fixed stepsize and CBCD belong to our framework, and any clustering point of the iterates generated by the proposed framework is a first-order stationary point. Preliminary experiments illustrate that our new framework is of great potential.

Key words. orthogonal constraint, Stiefel manifold, feasible method, trust-region subproblem, Householder transformation, block coordinate descent

1 Introduction

We consider numerical methods for solving the following matrix variable optimization problem

\[
\min_{X \in \mathbb{R}^{n \times p}} f(X) \quad \text{s.t.} \quad X^\top X = I_p,
\]

where \(I_p\) is the \(p\)-by-\(p\) identity matrix, \(f : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}\), and \(p < n\). Let \(S_{n,m} = \{Y \in \mathbb{R}^{n \times m} | Y^\top Y = I\}\). The feasible region of problem (1) can be simply denoted as \(S_{n,p}\).

The orthogonal constrained optimization model has applications in many areas, such as scientific engineering computing, data science and so on. More specifically, it plays an important role in electronic structure calculation [42, 43, 41], linear eigenvalue problems [6], low-rank correlation matrix problem [15], sparse principal component analysis [47, 8], orthogonal Procrustes problem [33, 11] and so on. For other applications, we refer the interested readers to references [10, 40, 16].

Throughout this paper, the following assumption is imposed to the objective function \(f\).

Assumption 1.1. (Blanket Assumption)
(i) \( f \) is twice differentiable with \( \rho \) to be the supremum of the 2-norm of its Hessian over \( \tilde{S} := \{ Y \mid \| Y \|_F^2 < p+1 \} \), i.e.,
\[
\rho := \sup_{X \in \tilde{S}} \| \nabla^2 f(X) \|_2.
\]

(ii) \( f(X) \) can be represented as \( h(X) + \text{tr}(G^T X) \), where \( G \in \mathbb{R}^{n \times p} \), and \( h(X) \) is orthogonal invariant, namely, \( h(XQ) = h(X) \) holds for any \( Q \in S_{p,p} \) and \( \nabla h(X) = H(X)X \), where \( H : \mathbb{R}^{n \times p} \to \mathbb{S}^n \) is a matrix function.

**Remark 1.2.** If \( \rho = 0 \), the objective function \( f(X) \) turns to be a linear function. In this case, the solution of (1) has the closed form \( X = -RQ^\top \), where \( RSQ^\top \) is the reduced singular value decomposition\(^2\) of \( G \). In this paper, this special situation will not be discussed.

Assumption 1.1 is required by the global convergence analysis of our algorithm framework. In this paper, we will not investigate how to avoid such restriction. Fortunately, the problems which we are interested in satisfy this assumption. Here are two simple examples.

**Example 1.1.**
\[
f(X) := \frac{1}{2} \text{tr}(X^\top AX) + \text{tr}(G^T X),
\]
where \( A \in \mathbb{S}^n \). In this case
\[
\nabla f(X) = AX + G.
\]

We notice that if the objective function defined in Example 1.1 takes \( G = 0 \), the corresponding orthogonal constrained optimization problem (1) reduces to the Rayleigh-Ritz minimization which is exactly the optimization model for the eigenvalue problem. However, the problem becomes difficult to solve if \( G \neq 0 \), even \( A \) is positive definite. Example 1.1 is also an extension of trust-region subproblem (see [43]). Therefore, to explore efficient solvers for this problem is already challenging and interesting.

**Example 1.2.**
\[
f(X) := \frac{1}{2} \text{tr}(X^\top AX) + \frac{1}{2} \sum_{i=1}^{m} q_i(z),
\]
where \( z = \text{diag}(XX^\top) \), \( q_i : \mathbb{R}^n \to \mathbb{R} \ (i = 1, \cdots, m) \) and \( A \in \mathbb{S}^n \). In this case,
\[
\nabla f(X) = \left( A + \sum_{i=1}^{m} \text{Diag}(\nabla q_i(z)) \right) X.
\]

This example often appears in electronic structure calculation [21], which is one of the most important topics in material science.

### 1.1 Overview of Existing Methods

In general, it is difficult to solve problem (1) to the global optimality due to the nonconvexity. In fact, finding a stationary point or even keeping feasibility is not an easy task because it can be numerically expensive to maintain the orthogonality for large \( p \). There are some existing infeasible methods such as the splitting method for orthogonal

---

\(^1\)In fact, \( \tilde{S} \) can be defined as any given bounded open set containing \( S_{n,p} \).

\(^2\)For \( G \in \mathbb{R}^{n \times p} \) with \( p < n \), the reduced singular value decomposition refers to \( RSQ^\top \) where \( R \in \mathbb{R}^{n \times p} \) and \( Q \in \mathbb{R}^{p \times p} \) are orthogonal matrices and \( S \in \mathbb{R}^{p \times p} \) is diagonal.
constrained problems [18] or the penalty method for large-scale eigenspace computation [39]. However, the former does not guarantee global convergence, and the latter only works for a very special case. Exploring practically useful infeasible methods for orthogonal constrained optimization problems is beyond the discussion of this paper.

Recently, there are some algorithms developed for special cases of (1), such as [46, 38] for electronic structure calculation, [23, 24] for dominant eigenpair calculation, [12] for computing the coupling between matrices, and so on. Usually, these approaches utilize the special structures of the problems and can hardly be extended to the generic orthogonal constrained optimization problems.

The feasible region of problem (1), $S_{n,p}$, is usually called Stiefel manifold [35]. Various optimization methods designed for solving optimization problems restricted on matrix manifold can be applied to problem (1). For instance, gradient based methods [25, 27, 1], conjugate gradient methods [10, 2], trust region methods [43], Newton methods [10], Quasi-Newton methods [32] and so on. The key principle of these methods is to find a feasible point with lower function value than the current iterate. In [10, 3], the authors study the geometric structure of Stiefel manifold from the optimization point of view, and bring up a new concept, which is called “retraction”, to connect previously unrelated algorithms. A map $R_X : T_X S_{n,p} \rightarrow S_{n,p}$ is called a retraction if the following properties hold.

1. $R_X(0_X) = X$, where $0_X$ is the origin of $T_X S_{n,p}$;
2. $\frac{d}{dt} R_X(tZ)|_{t=0} = Z$ for all $Z \in T_X S_{n,p}$,

where $T_X S_{n,p} := \{ Y \in \mathbb{R}^{n \times p} \mid Y^\top X + X^\top Y = 0 \}$ is the tangent space of Stiefel manifold $S_{n,p}$ at point $X$. The retraction $R_X$ maps a tangent vector into the manifold, so it defines an update rule to preserve the orthogonality.

There are two major classes of retraction for orthogonal constrained optimization problems. The first one searches along the geodesic of a manifold to find a suitable trial point. Methods in this class are called geodesic-like retractions [10, 1, 3]. Calculating geodesics involves solving ordinary differential equation, and hence exponential operations, which often cause computational difficulties, are inevitable. The authors of [27] propose a quasi-geodesic updating formula based on the Cayley transformation whose main computation is to solve an $n$-by-$n$ linear system. The methods in the other major class consist of two steps, line search in the tangent space and project back to the Stiefel manifold. Thus, they are called projection-like methods [25, 3, 4]. The orthogonal projection can be calculated by QR factorization, singular value decomposition (SVD) or polar decomposition. The projection-like methods coincide with the geodesic-like methods, in the special case of $p = 1$. The above mentioned retraction based approaches, including both geodesic-like and projection-like methods, should work with certain line search strategy, such as the Armijo inexact line search [28, 36] or nonmonotonic line search strategy. The line search procedure is to guarantee the global convergence, but at the meantime, it induces additional function value evaluations.

Recently, Wen and Yin [40] propose a feasible method for optimization with orthogonality constraints. In their work, an efficient way to calculate Cayley transformation is introduced. In each iteration, it only requires to solve a $2p \times 2p$ linear system instead of an $n \times n$ one. Combining a curvilinear search algorithms [13] with Barzilai-Borwein (BB) [5] nonmonotonic line search [45], it achieves much lower computational cost than the other existing retraction based algorithms and is illustrated to have robust numerical performance in solving a bunch of orthogonal constrained optimization problems. Later on, Jiang and Dai [16] significantly extend the idea of [40], and find out that a large group of retraction based methods enjoy such reducible iterative formulation. It can be proved that all the algorithms under their framework with BB nonmonotonic line search are globally convergent to a stationary point.

In order to clarify the difference among the aforementioned retraction based algorithms, we demonstrate their relationship through Figure 1.

It is worthy of mentioning that the retraction based algorithms highly depend on the geometry of the Stiefel manifold and hence have very low compatibility with additional constraints such as nonnegative constraints or linear
inequality constraints.

1.2 Contributions

In this paper, we revisit the first-order optimality condition of problem (1), and find it is of the following form,

\[
\begin{align*}
(I - XX^T)\nabla f(X) &= 0; & \text{stationarity} \\
X^T\nabla f(X) &= \nabla f(X)^T X; & \text{symmetry} \\
X^T X &= I_p. & \text{feasibility}
\end{align*}
\]

(2)

For convenience, we call the three equalities of (2) as stationarity, symmetry and feasibility, respectively. Based on the first-order optimality condition, we propose a new algorithm framework consisting of two main steps.

The first step is a function value reduction. Namely, we find a feasible point which reduces the objective function value to a certain amount in proportion to the violation of stationarity. We then propose two types of algorithms which can achieve such requirement. Gradient reflection (GR) and gradient projection (GP) are the representatives of the first type of algorithms, which uses different strategies to pull a gradient descent point back to the Stiefel manifold. The second type of algorithms employs a column-wise block coordinate descent (CBCD) iteration. A novel idea of solving the corresponding subproblem efficiently is proposed.

The second step is to find a feasible point satisfying the symmetry property. This correction step, whose main calculation is a $p \times p$ singular value decomposition, is highly dependent on Assumption 1.1. The correction step can be viewed as a rotation of the trial point obtained in the first step. In the special cases when $p = 1$ or $G = 0$, the symmetry of (2) always holds and hence this step can be waived.

Since our algorithm framework is not retraction based, it is more compatible with additional non-manifold constraints than retraction based approaches. Our framework exposes the essential mechanism of the gradient methods for orthogonal constrained optimization problems and hence global convergent gradient based approaches with a fixed stepsize can be developed. Moreover, as we show in the numerical experiments, in solving a class of generic quadratic minimization problems and the instances arising from electronic structure calculations, the implementations of our new algorithm framework perform robustly and more efficiently than the existing algorithms.

Finally, the global convergence of CBCD is of great potential itself, as this is the first convergence result for BCD method in solving nonconvex optimization problems with coupled constraints.
1.3 Organization

The rest of this paper is organized as follows. In Section 2, we study the first-order optimality condition of problem (1), and provides a new first-order framework. The main step of our new framework is only required to meet a sufficient function value reduction condition. We then develop two types of algorithms, in Section 3, to achieve this requirement and form three concrete algorithms under the scheme of the new framework, namely, GR, GP and CBCD, respectively. Global convergence results of our new algorithm framework are established in Section 4. In Section 5, we demonstrate the efficiency of our algorithm framework in solving a class of general quadratic minimization problems and the energy minimization problem arising from the electronic structure calculations. We show the great potential of our proposed approach in solving large-scale problems. Finally, conclusion remarks are given in the last section.

1.4 Notations

The set of n-by-n symmetric matrices and the n-by-n identity matrix are denoted by $\mathbb{S}^n$ and $I_n$, respectively. Euclidean inner product of two matrices $X, Y \in \mathbb{R}^{n \times p}$ is defined as $\langle X, Y \rangle = \text{tr}(X^\top Y)$, where $\text{tr}(A)$ is the trace of a matrix $A \in \mathbb{R}^{n \times n}$. $\|\cdot\|_2$ and $\|\cdot\|_F$ represent the 2-norm and F-norm, respectively. The notations $\text{diag}(A)$ and $\text{Diag}(x)$ stand for the vector formed by the diagonal entries of matrix $A$, and the diagonal matrix with the entries of $x \in \mathbb{R}^n$ to be its diagonal, respectively. $X^\dagger$ refers to the pseudo-inverse of $X$. We denote the smallest positive eigenvalue and the smallest eigenvalue in magnitude of $A$ by $\lambda_{\min}^+(A)$ and $\lambda_{\min}^-(A)$, respectively. The $i$-th column of matrix $X \in \mathbb{R}^{n \times p}$ is denoted by $X_i$. $X_i \in \mathbb{R}^{n \times (p-1)}$ denotes the matrix $X$ with its $i$-th column missed, i.e., $X_i = [X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_p]$. We use $X_{i,v} \in \mathbb{R}^{n \times p}$ to denote $X$ with its $i$-th column replaced by a given vector $v$, i.e., $X_{i,v} = [X_1, \ldots, X_{i-1}, v, X_{i+1}, \ldots, X_p]$. Finally, $B(C, r)$ is the ball defined as $\{X \in \mathbb{R}^{m_1 \times m_2} \mid \|X - C\|_F \leq r\}$, where $C \in \mathbb{R}^{m_1 \times m_2}$ is the center and $r$ is the radius. $\text{qr}(X)$ is the Q matrix of the reduced QR decomposition of $X$. $\mathcal{P}_{S_{n,p}}(X)$ denotes the projection to Stiefel manifold $S_{n,p}$. Finally, $\text{rand}(n, p)$ and $\text{randn}(n, p)$ represent $n \times p$ randomly generated matrices under i.i.d. uniformed distribution in $[0, 1]$ and i.i.d. standard Gaussian distribution, respectively.

2 A New First-order Framework

In this section, we first give a new presentation of the first-order optimality of the orthogonal constrained optimization problem (1), which motivates our new framework. The details of the new framework will also be presented.

2.1 Optimality Condition

The first-order optimality condition of problem (1) can be interpreted as follows.

**Definition 2.1.** Given a point $X \in \mathbb{R}^{n \times p}$, if the relationship

$$
\begin{cases}
\text{tr}(Y^\top \nabla f(X)) \geq 0; \\
X^\top X = I_p
\end{cases}
$$

holds for any $Y \in \mathcal{T}_X S_{n,p}$, we call $X$ a first-order stationary point of (1). The set containing all the first-order stationary point is simply denoted as $\Omega_{FON}$.

---

3 $Q \in \mathbb{R}^{n \times p}$ is the Q matrix of the reduced QR decomposition of $X \in \mathbb{R}^{n \times p}$, if $X = QR$, $Q \in \mathbb{R}^{n \times p}$ is orthogonal and $R \in \mathbb{R}^{p \times p}$ is an upper triangle matrix.

4 $\mathcal{P}_{S_{n,p}}(X) = UV^\top$ where $\tilde{U} \Sigma \tilde{V}^\top$ is the reduced singular value decomposition of $X$. 

---

5
Since the above condition is not computationally checkable, we propose the following equivalent form.

**Lemma 2.1.** A point $X$ is a first-order stationary point if and only if equalities (2) hold.

**Proof.** We notice that any $Y \in T_X S_{n,p}$ can be uniquely decomposed as $Y = XS + K$, where $S \in \mathbb{R}^{p \times p}$ is a skew matrix (i.e., $S^T + S = 0$) and $K \in \mathbb{R}^{n \times p}$ satisfies $K^TX = 0$, which is equivalent to $K = (I_n \!-\! XX^T)K$.

Since $S$ and $K$ are arbitrary, condition (3) is equivalent to the following relationships

$$\begin{align*}
\text{tr}(S^TX^T\nabla f(X)) &\geq 0, \quad \forall S \in \mathbb{R}^{p \times p} \text{ and } S^T + S = I_p, \\
\text{tr}(K^T\nabla f(X)) &\geq 0, \quad \forall K \in \mathbb{R}^{n \times p} \text{ and } K^TX = 0,
\end{align*}$$

By using (4) and the skew symmetry of $Q^T - Q$, where $Q := X^T \nabla f(X)$, we obtain

$$\text{tr}((Q - Q^T)Q) \geq 0. \quad (6)$$

It then follows from (6) that

$$0 \leq \text{tr}((Q - Q^T)Q) + \text{tr}((Q - Q^T)Q) = \text{tr}(QQ - Q^TQ) + \text{tr}(Q^T(Q^T - Q)) = \text{tr}(QQ - Q^TQ + Q^TTQ - QQ^T) = \text{tr}((Q - Q^T)(Q - Q^T)) = -\text{tr}((Q - Q^T)^T(Q - Q^T)) \leq 0.$$

The equality only holds when $Q = Q^T$. On the other hand, if $X^T \nabla f(X)$ is symmetric, the equality $\text{tr}(S^TX^T\nabla f(X)) = 0$ holds for any skew symmetric matrix $S$. Hence, (4) is equivalent to the symmetry of $X^T \nabla f(X)$.

Following from the property $K = (I_n \!-\! XX^T)K$ and the arbitrariness of $K$, we can easily obtain the equivalence between (5) and $(I_n \!-\! XX^T)\nabla f(X) = 0$. This completes the proof. \hfill \Box

**Remark 2.2.** It is very easy to check that our first-order optimality condition (2), considered in Euclidean space, is exactly the same as the one in tangent space

$$\begin{cases}
\nabla f(X) - XX^T \nabla f(X)^T X = 0; \\
X^T X = I_p,
\end{cases}$$

which is stated in [40] etc. Moreover, it actually holds that

$$\|\nabla f(X) - XX^T \nabla f(X)^T X\|_F^2 = \|\nabla f(X) - XX^T \nabla f(X)\|_F^2 + \|X^T \nabla f(X) - \nabla f(X)^T X\|_F^2. \quad (7)$$

### 2.2 Symmetrization Step and Algorithm Framework

We notice that there are three properties, stationarity, symmetry and feasibility in our first-order optimality condition (2) of problem (1). Motivated by the relationship (7), to make the gradient in tangent space equal to zero, we can adopt the following two steps procedure. From the current iterate, we first find a trial point which reduces the function value in proportion to the violation of the stationarity. Based on this trial point, we then find next iterate which makes the symmetry property hold without increasing the function value. Then we repeat the procedure till converging. In these two steps, the feasibility holds all the time. The details of these two steps are described as the following.
Lemma 2.3. Suppose \( X \) holds that
\[
\bar{X} = (SVD) \text{ for a } p \times p \text{ matrix}
\]
It suffices to consider the symmetrization of \( \bar{X} \). In the second part, we consider to construct a correction step which makes the symmetry property hold without increasing the function value.

Proof. The orthogonality of \( \theta \) and the truncation of \( \bar{X} \) can be directly derived by the formula (11). Next, we try to prove inequality (12). If \( \theta = 0 \), which means \( \nabla f(X) = H(X)X \). Then, the symmetry of \( \bar{X} \) implies (12) immediately. On the other hand, according to Assumption 1.1, we have
\[
f(\bar{X}) = f(\bar{X}) - f(\bar{X}) = h(\bar{X}) + \text{tr}(G^T \bar{X}) - h(X) - \text{tr}(G^T X) = \text{tr}(G^T \bar{X} - G^T X)
\]

where \( B = (\Lambda^T U + U^T T\Lambda)/2 \).

On the other hand,
\[
||\bar{X} - G^T X||^2_F = ||\bar{X} - G^T X||^2_F
\]

where \( C_1 > 0 \) is a positive constant. The right hand side of (8) measures the square of the F-norm of the projected gradient at \( X \) in the Euclidean space.

Although the intermediate point \( \bar{X} \) achieves a sufficient function value reduction and is orthogonal, it does not satisfy the symmetry property in (2). In the second part, we consider to construct a correction step which makes the symmetry property hold without increasing the function value.
where the last equality uses the fact that
\[ \text{tr}(B^2) = \frac{1}{2}\text{tr}(A^2) + \frac{1}{2}\text{tr}(AT^TUA^T U). \]

Moreover, we have
\[
\text{tr}(A^2 - B^2) \leq \sum_{i=1}^{p} (A_{ii}^2 - B_{ii}^2) \leq \sum_{i=1}^{p} (A_{ii}^2 - B_{ii}^2)(A_{ii} + B_{ii}) \\
\leq 2 \sum_{i=1}^{p} \Lambda_{ii}(A_{ii} + B_{ii}) \leq 2||\Lambda||_2 \cdot \sum_{i=1}^{p} (A_{ii} + B_{ii}) = 2||\Lambda||_2 \cdot \text{tr}(A + B) \\
\leq 2||G||_2 \cdot \text{tr}(A + B) = 2\theta \cdot \text{tr}(A + B).
\]

Here the third inequality uses the fact that
\[ |B_{ii}| = \Lambda_{ii} \cdot |T_i^TU_i| \leq \Lambda_{ii}. \]

Combining (13)-(15), we complete the proof.

We can adopt \( c(X) := (I_n - XX^T)\nabla f(X) \) for our stop criterion. This is because the symmetry and the feasibility of (2) hold at each iteration. The complete framework can be described as the following.

\begin{algorithm}
\caption{First-order Framework for Orthogonal Constrained Minimization}
\begin{algorithmic}[1]
\State Set tolerance \( \epsilon > 0 \); Initialize: \( X^0 \in S_{n,p} \); Set \( k := 0 \)
\While {\( ||c(X^k)||_F > \epsilon \)}
\State Based on \( X^k \), find a feasible point \( \bar{X} \) satisfying (8);
\State Based on \( \bar{X} \), calculate a feasible point \( X^{k+1} \) by (11);
\State Set \( k := k + 1 \).
\EndWhile
\State Return \( X^k \).
\end{algorithmic}
\end{algorithm}

3 Algorithms for Finding \( \bar{X} \) from \( X^k \)

In Section 2, we propose a new algorithm framework, however, how to find a point \( \bar{X} \) satisfying the sufficient function value reduction property (8) is still open. In this section, we introduce two types of algorithms to achieve Step 3 in Algorithm 1. The first type of algorithms is based on gradient descent in the Euclidean space which will be introduced in the first two subsections. The second type of algorithms adopts a column-wise coordinate descent idea and it will be introduced in the third subsection. In the last subsection, we list the computational cost per iteration of some existing algorithms and our new proposed algorithms.

3.1 Gradient Type Method

An intuitive idea to reduce the function value in the Euclidean space is to take the gradient descent direction. Unfortunately, a trial point obtained by a gradient descent step from the current iterate must violate the orthogonal constraint. Therefore, in this section we discuss two concrete strategies to pull the trial point back to the Stiefel manifold. Each of them can be an option of Step 3 in Algorithm 1.
Both of these two strategies are based on the following observation.

**Lemma 3.1.** For any \( Y \in B_{X, \tau} := B(X - \tau \nabla f(X), \tau \| \nabla f(X) \|_F) \), where \( \tau \in (0, \rho^{-1}) \), it holds that

\[
 f(X) - f(Y) \geq \frac{1 - \rho \tau}{2\tau} \cdot \| X - Y \|_F^2.
\]  

**Proof.** For any \( Y \in B_{X, \tau} \), we can derive

\[
\langle Y - X, Y - X + 2\tau \nabla f(X) \rangle \leq 0,
\]

which implies

\[
f(Y) \leq f(X) + \langle Y - X, \nabla f(X) \rangle + \frac{\rho}{2} \| Y - X \|_F^2
\]

\[
= f(X) + \frac{1}{2\tau} \cdot \langle Y - X, Y - X + 2\tau \nabla f(X) \rangle - \frac{\tau^{-1} - \rho}{2} \cdot \| Y - X \|_F^2
\]

\[
\leq f(X) - \frac{\tau^{-1} - \rho}{2} \cdot \| Y - X \|_F^2.
\]

This completes the proof. \( \square \)

We illustrate the relationship among the feasible region, current iterate, gradient step and the auxiliary ball \( B_{X, \tau} \) in Figure 2.

![Figure 2: Gradient Type Method](image)

3.1.1 Gradient Reflection

The first possible choice of a feasible trial point is to take the reflecting point of the current iterate \( X^k \) with respect to the function value reduction point \( X - \tau \nabla f(X) \). This point can be actually calculated by the Householder transformation

\[
GR : \begin{cases} 
V = X^k - \tau \cdot \nabla f(X^k), & \forall \tau \in (0, \rho^{-1}); \\
\bar{X} = (-I_n + 2V(V^\top V)^{-1}V^\top)X^k,
\end{cases}
\]  

because \( \bar{X} \) is the reflection point of \( X \) across an infeasible gradient descent step \( V \). As a consequence, we call the algorithm based on our new framework Algorithm 1 with (17) used in Step 3 the Gradient Reflection (GR).
Next, we show the intermediate point $\bar{X}$ defined in (17) is feasible and achieves a sufficient function value reduction (8).

**Lemma 3.2.** Let $X^k \in S_{n,p}$ and $\bar{X}$ be defined by (17). Then it holds that $\bar{X} \in S_{n,p}$ and

$$f(X^k) - f(\bar{X}) \geq \frac{2(\tau^{-1} - \rho)}{(\tau^{-1} + \rho + \theta)^2} \cdot \| (I_n - X^k X^{k^\top}) \nabla f(X^k) \|^2_F,$$

where $\tau \in (0, \rho^{-1})$.

**Proof.** With a slight abuse of notation, we omit the superscript $k$ and use $X$ to denote $X^k$ in this proof.

First, by simple calculation, we have $\bar{X}^\top \bar{X} = X^\top (-I_n + 2V(V^\top V)^\top V^\top) (-I_n + 2V(V^\top V)^\top V^\top) X = I_p$. Namely, $\bar{X} \in S_{n,p}$.

Denoting $RSQ^\top$ to be the reduced singular value decomposition of $V$, we have

$$\| \bar{X} - X \|_F = 2 \| (I_n - V(V^\top V)^\top V^\top) X \|_F = 2 \| (I_n - RR^\top) X \|_F = 2 \| (I_n - XX^\top) R \|_F$$

$$= 2 \| (I_n - XX^\top) VQ S^\top \|_F \geq 2 \| (I_n - XX^\top) VQ \|_F \cdot \lambda_{\text{min}}(S^\top)$$

$$= 2 \| (I_n - XX^\top) V \|_F / \| S \|_2 \geq 2 \tau \| c(X) \|_F / \| V \|_2 \geq \frac{2}{\tau^{-1} + \rho + \theta} \| c(X) \|_F.$$  \hspace{1cm} (19)

Here, the last inequality results from

$$\| \nabla f(X) \|_2 \leq \| H(X) X \|_2 + \| G \|_2 \leq \rho + \theta,$$

$$\| V \|_2 \leq \| X \|_2 + \| \nabla f(X) \|_2 \leq 1 + \tau (\rho + \theta).$$

Substituting the inequality (19) into (16) in Lemma 3.1 with $Y = \bar{X}$, we arrive at

$$f(X) - f(\bar{X}) \geq \frac{4}{(\tau^{-1} + \rho + \theta)^2} \cdot \frac{\tau^{-1} - \rho}{2} \cdot \| (I_n - XX^\top) \nabla f(X) \|^2_F$$

$$= \frac{2(\tau^{-1} - \rho)}{(\tau^{-1} + \rho + \theta)^2} \cdot \| (I_n - XX^\top) \nabla f(X) \|^2_F,$$

which completes the proof. \hfill \Box

### 3.2 Gradient Projection

Another possible choice of a feasible trial point is to directly take the projection of $X - \tau \nabla f(X)$ onto the Stiefel manifold, which can be calculated by,

$$\text{GP : } \begin{cases} V = X^k - \tau \cdot \nabla f(X^k), & \forall \tau \in (0, \rho^{-1}); \\ \bar{X} = P_{S_{n,p}}(V). \end{cases}$$ \hspace{1cm} (21)

We call the algorithm based on our new framework Algorithm 1 with (21) used in Step 3 the Gradient Projection (GP). We can similarly prove the feasibility of $\bar{X}$ and show the sufficient function value reduction can be achieved.

**Lemma 3.3.** Let $X^k \in S_{n,p}$ and $\bar{X}$ be defined by (21). Then it holds that $\bar{X} \in S_{n,p}$ and

$$f(X^k) - f(\bar{X}) \geq \frac{\tau^{-1} - \rho}{2(\tau^{-1} + \rho + \theta)^2} \cdot \| (I_n - X^k X^{k^\top}) \nabla f(X^k) \|^2_F,$$ \hspace{1cm} (22)
where $\tau \in (0, \rho^{-1})$.

**Proof.** With a slight abuse of notation, we omit the superscript $k$ and use $X$ to denote $X^k$ in this proof, and denote $\bar{X}_{GR}$ as the trial point obtained by (17).

We first obtain

$$
||\bar{X} - X||_F^2 - \frac{1}{4}||\bar{X}_{GR} - X||_F^2 = \left\|\bar{U} \bar{Z}^T - X\right\|_F^2 - \left\|(I_n - X_{GR}X_{GR}^T)\bar{U}\right\|_F^2 = \left\|X - \bar{U} \bar{Z}^T\right\|_F^2 \geq 0,
$$

which implies

$$
||\bar{X} - X||_F \geq \frac{1}{2}||\bar{X}_{GR} - X||_F.
$$

Here $\bar{U}\Sigma\bar{Z}^T$ is the reduced singular value decomposition of $V$. Then we can complete the proof by following the same manner as Lemma 3.2.

**Remark 3.4.** It is worthy of mentioning that the QR decomposition can not be used to pull the gradient step in Euclidean space back to the feasible region. Namely, the following updating formula can not guarantee a feasible trial point that achieves the sufficient function value reduction.

$$
\begin{align*}
V_\tau &= X^k - \tau \cdot \nabla f(X^k), \quad \forall \tau > 0 \\
\bar{X}_\tau^\pm &= \pm \tilde{Q},
\end{align*}
$$

(23)

where $V_\tau = \tilde{Q} \tilde{R}$ is the unique reduced QR decomposition of $V_\tau$. More specifically, we can illustrate it by the following example.

**Example 3.1.** Suppose $f(X) = \frac{1}{2} \text{tr}(X^TAX) + \text{tr}(G^TX)$, where $A = \begin{bmatrix} 4 & 0 & -2 & 0 \\ 0 & 4 & 0 & -2 \\ -2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix}$, $G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

Let $X^k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $\bar{X}_\tau^\pm$ be defined by (23). It holds that

$$
X^k \in S_{n,p}, \quad X^k \nabla f(X^k) = \nabla f(X^k)^T X^k,
$$

$$
f(X^k) - f(\bar{X}_\tau^\pm) < 0, \quad \forall \tau > 0.
$$

**Proof.** The $X^k$ part can be verified easily. We give a rough proof of the second part. Moreover, we have $f(X^k) = 0$. The unique reduced QR decomposition of $V_\tau$ is

$$
V_\tau = \tilde{Q} \tilde{R} = \begin{bmatrix} -\tau/\sqrt{1 + \tau^2} & 0 \\ 0 & -\tau/\sqrt{1 + \tau^2} \\ 1/\sqrt{1 + \tau^2} & 0 \\ 0 & 1/\sqrt{1 + \tau^2} \end{bmatrix} \begin{bmatrix} \sqrt{1 + \tau^2} & 0 \\ 0 & \sqrt{1 + \tau^2} \end{bmatrix}.
$$
Hence, 
\[ f(\bar{X}_{\tau}^\pm) = \frac{4\tau^2 + 4\tau \pm 2\tau \sqrt{1 + \tau^2}}{1 + \tau^2} > 0, \quad \forall \tau > 0, \] 
which completes the proof. \(\square\)

In fact, it can also be shown that \(\bar{X}_{\tau}^\pm\) is not in the auxiliary ball \(B_{\bar{X},\tau}\) defined in Lemma 3.1.

### 3.3 Column-wise Block Coordinate Descent Method

In this subsection, we employ a column-wise block coordinate descent (CBCD) approach for Step 3 in Algorithm 1. Once we fix the values of \(p - 1\) columns of \(X\) and only leave the \(i\)-th column as variable, we arrive at the following subproblem,

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f_{i,X}(x) \\
\text{s.t.} & \quad ||x||_2 = 1, \\
& \quad X_i^\top x = 0, \\
\end{align*}
\]

where \(f_{i,X}(x) := f(X_{i,x})\), \(X_{i,x}\) and \(X_i\) are defined in Subsection 1.4.

Suppose we can obtain the solution of the above subproblem or find a feasible point \(x^+\) with a sufficient function value reduction comparing with \(f_{i,X}(X_i)\). Then we can use this feasible point to update our iterate column-wisely in a Gauss-Seidel manner. More specifically, if \(X\) is the current iterate, the trial point \(\bar{X}\) can be calculated by the following CBCD scheme.

**Algorithm 2:** Column-wise Block Coordinate Descent

1. Set \(W^0 = X, i := 1;\)
2. while \(i \leq p\) do
3. \quad Solve the subproblem (24) with \(X_i\) replaced by \(W_{i-1}^i\), and obtain feasible point \(x^+\) satisfying the following sufficient function value reduction and asymptotic small stepsize safe guard
   \[
   f_{i,W_{i-1}^i}(X_i) - f_{i,W_{i-1}^i}(x^+) \geq k_1 ||X_i - x^+||^2_2, \quad (25)
   \]
   \[
   ||X_i - x^+||_2 \geq k_2 (I - W_{i-1}^iW_{i-1}^i\top) \nabla f_{i,W_{i-1}^i}(X_i)||_2; \quad (26)
   \]
   \quad Set \(W^i = W_{i-1}^i, i := i + 1;\)
4. Return \(\bar{X} = W^p.\)

**Remark 3.5.** Algorithm 2 actually provides a cyclic column-wise block coordinate descent scheme, i.e. the columns are updated in a cyclic order. We can similarly implement the greedy order, stochastic order (sampling with replacement), or randomly permuted order (sampling without replacement) which often appear in classical block coordinate descent algorithms. However, as we will show in Section 5, these strategies will not help to improve the performance of cyclic CBCD. Therefore, we omit the detailed descriptions and analysis of these strategies.

Before claiming Algorithm 2 can find \(\bar{X}\) in Step 3 of Algorithm 1, we need to answer two questions: can we cheaply calculate a solution or feasible point achieving sufficient function value reduction and asymptotic small step-size safe guard (25)-(26)? Does Algorithm 2 provide a feasible point of problem (1) satisfying (8)? We answer these two questions in the following two subsections.
3.3.1 Solving the CBCD Subproblem

In this subsection, we discuss how to obtain a feasible trial point of subproblem (24) efficiently. We notice that the second constraint of (24) restricts the variable $x$ lying in the null space of $X_i$. Hence, we can use the variable change $x = (I - X_iX_i^\top)x$ to reduce this constraint.

**Proposition 3.6.** Subproblem (24) is equivalent to the following problem

$$
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f_i(x) := f_i,\mathcal{X}(I - X_iX_i^\top)x \\
\text{s.t.} & \quad \|X_i X_i^\top x\|_2 = 1.
\end{align*}
$$

The fact $X_i^\top x = 0$ holds if and only if $x = (I - X_iX_i^\top)x$. Hence, we can obtain Proposition 3.6 by simple substitution and hence omit the proof. Although problem (27) is still difficult to tackle due to the singularity of the constraint of (27), it is equivalent to a well-posed problem if restricted to a subspace orthogonal to $X_i$.

**Proposition 3.7.** The equivalence between problem (27) restricted to a subspace $\mathcal{D}$ and the following sphere constrained problem

$$
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad q_i(x) := f_i,\mathcal{X}(I - X_iX_i^\top)x \\
\text{s.t.} & \quad \|x\|_2 = 1, \\
& \quad x \in \mathcal{D}
\end{align*}
$$

holds, if the orthogonality $X_i^\top x = 0$ is satisfied by any $x \in \mathcal{D}$.

**Proof.** For any $x \in \mathcal{D}$, it holds that $x = (I - X_iX_i^\top)x$ which implies the equivalence. \hfill \Box

Propositions 3.6 and 3.7 tell us that we can calculate a feasible point of subproblem (24) with sufficient function value reduction through solving problem (28) if we can find a suitable subspace $\mathcal{D}$.

We notice that both $X_i$ and $\nabla q_i(X_i) = (I - X_iX_i^\top)\nabla f_i,\mathcal{X}(I - X_iX_i^\top)X_i$ lie in the null space of $X_i$. Therefore, any point in the subspace $\text{span} \{X_i, \nabla q_i(X_i)\}$ satisfies the orthogonality. Namely, $\text{span} \{X_i, \nabla q_i(X_i)\}$ is a qualified choice of orthogonal subspace $\mathcal{D}$ in Proposition 3.7. Considering that subproblem (28) with $\mathcal{D} = \text{span} \{X_i, \nabla q_i(X_i)\}$ is a special case of the original orthogonal constrained optimization problem (1) in the special case $n = 2$ and $p = 1$, we recommend to use the GR step (17) or the GP step (21) introduced in Subsection 3.1 to calculate $x^+$.

It can be verified that either GR or GP step satisfies sufficient function value reduction (25) and asymptotic small stepsize safe guard (26).

**Lemma 3.8.** Let $x^+ = (-1 + 2v(v^Tv)^{-1}v)X_i$ or $x^+ = (v^Tv)^{-\frac{1}{2}}v$, where $v = X_i - \tau \cdot \nabla q_i(X_i)$, $\tau \in (0, \rho^{-1})$. Then $x^+$ satisfies the constraints of (24) and conditions (25) and (26).

The proof of Lemma 3.8 directly follows from Lemmas 3.1, 3.2, 3.3 and the fact that $I - XX^\top = (I - X_iX_i^\top)(I - X_iX_i^\top)$, and hence it is omitted here.

**Remark 3.9.** If $f_i,\mathcal{X}$ is quadratic, subproblem (28) restricted to the subspace $\text{span} \{X_i, \nabla q_i(X_i)\}$ is equivalent to finding the roots of a quartic equation, which can be calculated in closed form. In this case, the global minimizer of subproblem (28) restricted to the subspace $\text{span} \{X_i, \nabla q_i(X_i)\}$ can be an alternative option of $x^+$. 


3.3.2 Sufficient Function Value Reduction

In this subsection, we show that \( \bar{X} \) calculated by Algorithm 2 is a feasible point of problem (1) and satisfies the sufficient function value reduction property (8).

Lemma 3.10. Let \( X \in S_{n,p} \) and \( \bar{X} \) be calculated by Algorithm 2. Then it holds that \( \bar{X} \in S_{n,p} \) and

\[
 f(X) - f(\bar{X}) \geq \frac{k_1 k_2^2}{(1 + (p - 1)k_1 k_2 ((1 + \sqrt{2})\rho + \sqrt{2}\theta))^2} \cdot \| (I_n - XX^T) \nabla f(X) \|^2_F.
\]  

(29)

Proof. The feasibility of \( \bar{X} \) directly follows from the cyclic Gauss-Seidel type update and the constraints of subproblem (24).

Now, we prove the second part. First, we have

\[
 f(X) - f(\bar{X}) = f(W^0) - f(W^p) = \sum_{i=1}^p (f(W^{i-1}) - f(W^i)) = \sum_{i=1}^p (f_{i,W^{i-1}}(W_{i}^{i-1}) - f_{i,W^{i-1}}(W_{i}^i))
\]  

(30)

and

\[
 f_{i,W^{i-1}}(W_{i}^{i-1}) - f_{i,W^{i-1}}(W_{i}^i) \geq k_1 k_2^2 \| (I - W^{i-1}W^{i-1}^T) \nabla f_{i,W^{i-1}}(W_{i}^{i-1}) \|^2_F.
\]  

(31)

By using the facts \( \| W_{i}^{i-1} W_{i}^i \| \leq 1 \) and \( \sqrt{2 - 2\delta^2} \leq 2\sqrt{1 - \delta} (\forall \delta) \leq 1 \), the Lipschitz continuity and the boundedness of gradient (20), we have

\[
 \| (I - W^{i-1}W^{i-1}^T) \nabla f_{i,W^{i-1}}(W_{i}^{i-1}) \|_2 \\
\geq \| (I - W^0W^0^T) \nabla f_{i,W^{i-1}}(W_{i}^{i-1}) \|_2 - \| (W^{i-1}W^{i-1}^T - W^0W^0^T) \nabla f_{i,W^{i-1}}(W_{i}^{i-1}) \|_2 \\
\geq \| (I - XX^T) \nabla f_{i,W^{i-1}}(X_i) \|_2 - \sum_{j=1}^{i-1} \| W_j^TW_j^T - W_j^TW_j^T \|_F \cdot \| \nabla f_{i,W^{i-1}}(W_{i}^{i-1}) \|_2 \\
\geq \| (I - XX^T) \nabla f_{i,X}(X_i) \|_2 - \| \nabla f(W^0) - \nabla f(W^{i-1}) \|_F - \sqrt{2}(\rho + \theta) \cdot \sum_{j=1}^{i-1} \left( \sqrt{2 - 2 \left( W_j^TW_j^T \right)^2} \right) \\
\geq \| (I - XX^T) \nabla f_{i,X}(X_i) \|_2 - \rho \cdot \| W^0 - W^{i-1} \|_F - \sqrt{2}(\rho + \theta) \sum_{j=1}^{i-1} \| W_j^T - W_j^T \|_2 \\
\geq \| (I - XX^T) \nabla f_{i,X}(X_i) \|_2 - ((1 + \sqrt{2})\rho + \sqrt{2}\theta) \sqrt{k_1} \sum_{j=1}^{i-1} \sqrt{f_{j,W^{j-1}}(W_{j}^{j-1}) - f_{j,W^{j-1}}(W_{j}^j)}. 
\]

Together with (31), we have

\[
 \sqrt{k_1 k_2} \| (I - XX^T) \nabla f_{i,X}(X_i) \|_2 - k_1 k_2 ((1 + \sqrt{2})\rho + \sqrt{2}\theta) \sum_{j=1}^{i-1} \sqrt{f_{j,W^{j-1}}(W_{j}^{j-1}) - f_{j,W^{j-1}}(W_{j}^j)}. 
\]

(32)
Substituting (32) into (30) and using the fact that 
\[
(\delta_1 + c \sum_{j=1}^{s} \delta_j)^2 \leq (1 + sc)\delta_1^2 + \sum_{j=1}^{s} c(1 + sc)\delta_j^2,
\]
we have
\[
\left(1 + (p - 1)k_1k_2((1 + \sqrt{2})\rho + \sqrt{2}\theta)\right)^2 \sum_{i=1}^{p} \left(\sqrt{f_{i,W_1^{-1}}(W_i^{-1}) - f_{i,W_1^{-1}}(W_i^p)}\right)^2 \geq \sum_{i=1}^{p} k_1k_2^2 : \|f_{i,W_1^{-1}}(W_i^{-1}) - f_{i,W_1^{-1}}(W_i^p)\|^2,
\]
which implies
\[
f(X) - f(\bar{X}) \geq \frac{k_2}{1 + (p - 1)k_1k_2((1 + \sqrt{2})\rho + \sqrt{2}\theta)^2} \cdot \|(I - XX^\top)\nabla f(X)\|_F^2.
\]
This completes the proof. \qed

A byproduct of the proof of Lemma 3.10 is the following asymptotic small stepsize safeguard property of CBCD.

**Corollary 3.11.** Let \( X \in S_{n,p} \) and \( \bar{X} \) be calculated by Algorithm 2. Then it holds that
\[
\|(X - \bar{X})\|_F \geq \frac{k_2}{1 + (p - 1)k_1k_2((1 + \sqrt{2})\rho + \sqrt{2}\theta)^2} \cdot \|(I_n - XX^\top)\nabla f(X)\|_F.
\]

### 3.4 Computational Cost

In this subsection, we compare the computational cost per iteration among the existing algorithms and our GR, GP and CBCD. First of all, we clarify the computational cost of the basic linear algebra operations as the following. Given \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times p} \) and \( x \in \mathbb{R}^n \), calculating matrix-matrix product \( AB \) needs \( 2n^2p \) flops while computing matrix-matrix product \( B^\top B \) needs \( np^2 + np \) flops. Computing \( A^{-1} \) needs \( 8n^3/3 \) flops and computing \( Ax \) needs \( 2n^2 \) flops. We assume \( \nabla f(X) \) is already assembled and hence the computation of \( \nabla f(X) \) is not counted in the computational cost per iteration. The other settings are similar to Table 1 of [16]. We illustrate the comparison result as the following.

<table>
<thead>
<tr>
<th>Update schemes</th>
<th>Computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>first ( \tau )</td>
</tr>
<tr>
<td>geodesic-like algorithms</td>
<td></td>
</tr>
<tr>
<td>( Y_{geoc}(\tau; X) ) [10]</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>( Y_{geo}(\tau; X) ) [27]</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>( Y_{geod}(\tau; X) ) [10]</td>
<td>( 10np^2 + 2np + O(p^3) )</td>
</tr>
<tr>
<td>( Y_{geo}(\tau; X) ) [40]</td>
<td>( 7np^2 + 2np + O(p^3) )</td>
</tr>
<tr>
<td>projection-like algorithms</td>
<td></td>
</tr>
<tr>
<td>( Y_{p}(\tau; X) ) [3]</td>
<td>( 6np^2 + 3np + O(p^3) )</td>
</tr>
<tr>
<td>( Y_{q}(\tau; X) ) [10]</td>
<td>( 7np^2 + 4np + O(p^3) )</td>
</tr>
<tr>
<td>( Y_{q}(\tau; X) ) [25]</td>
<td>( 7np^2 + 4np + O(p^3) )</td>
</tr>
<tr>
<td>( Y_{q}(\tau; X) ) [16]</td>
<td>( 7np^2 + 3np + O(p^3) )</td>
</tr>
<tr>
<td>our algorithms</td>
<td></td>
</tr>
<tr>
<td>GR</td>
<td>( 9np^2 + 3np + O(p^3) )</td>
</tr>
<tr>
<td>GP</td>
<td>( 7np^2 + 2np + O(p^3) )</td>
</tr>
<tr>
<td>CBCD-GR</td>
<td>( 4np^2 + 8np + O(p^3) )</td>
</tr>
<tr>
<td>CBCD-GP</td>
<td>( 4np^2 + 5np + O(p^3) )</td>
</tr>
</tbody>
</table>

Table 1: Comparison on computational cost
In Table 1, the two columns “first $\tau$” and “subsequent $\tau$” refer to the computational cost for the first trial point, and for subsequent trial points, respectively. However, the additional function value evaluations have not been counted yet. For our GP, GR and CBCD, line search is waived, as GR and GP converge with a fixed stepsize and the subproblem of CBCD only needs to be solved inexactly by one iteration. Hence, our computational cost per iteration is much cheaper than the retraction based algorithms in general.

Moreover, CBCD-GR or CBCD-GP refers to the CBCD (using Algorithm 2 in the Step 3 of Algorithm 1) with GR or GP updating formula used once in Step 3 of Algorithm 2. We notice that the calculation of $\nabla f_i(X)$ is waived because it is equal to $\nabla f_i(X)$ which is implied by $W_i^{-1}(X_i) = 0$. If $f_i(X_i)$ are quadratic, and we solve the subproblem (28) restricted to the subspace span $\{X_i, \nabla q_i(X_i)\}$ to the global optimality in Algorithm 2, the corresponding computational cost is $12np^2 + 3np + O(p^3)$.

4 Convergence Analysis

In this section, we establish the global convergence of our new algorithm framework, Algorithm 1. First, the function value convergence is put forward.

**Lemma 4.1.** Let $\{X^k\}$ be the iterate sequence generated by Algorithm 1 initiated from a point $X^0 \in S_{n,p}$, then $\{f(X^k)\}$ converges.

**Proof.** According to the construction of $\bar{X}$ in Step 3 of Algorithm 1 and Lemma 2.3, we obtain

$$f(X^k) - f(X^{k+1}) \geq C_1 \|\nabla f(X^k) - X^k \nabla f(X^k)^T X^k\|_F^2.$$  \hspace{1cm} (34)

So $\{f(X^k)\}$ is monotonically decreasing. Since $S_{n,p}$ is a compact set, $\{f(X^k)\}$ is bounded below, we can conclude that $\{f(X^k)\}$ converges.

Next, we show the iterate subsequence convergence result.

**Theorem 4.2.** Let $\{X^k\}$ be the iterate sequence generated by Algorithm 1 initiated from a point $X^0 \in S_{n,p}$. Then there exists a convergent subsequence of $\{X^k\}$. Moreover, each accumulation point $X^*$ of $\{X^k\}$ satisfies the first-order optimality condition of (1), namely, it holds that $X^* \in \Omega_{FON}$.

**Proof.** Notice that $\{X^k\}$ is bounded and hence has a convergent subsequence. By combining Lemmas 2.1 and 4.1, the relationship (11), and the feasibility of each iterate, we complete the proof.

Lemma 4.1 and Theorem 4.2 guarantee that $f(X)$ is a constant on the accumulation point set of $\{X^k\}$. We denote this constant as $f^*$, and

$$\Omega_{FON}^{f^*} = \Omega_{FON} \cap \{X | f(X) = f^*\}.$$  \hspace{1cm} (35)

Next, we show that the distance between $X_k$ and $\Omega_{FON}^{f^*}$ goes to zero.

**Corollary 4.3.** Let $\{X^k\}$ be the iterate sequence generated by Algorithm 1 initiated from a point $X^0 \in S_{n,p}$, then it holds that

$$f(X^k) \geq f^*, \quad \forall k = 1, \cdots$$  \hspace{1cm} (36)
and
\[
\lim_{k \to \infty} \operatorname{dist}(X^k, \Omega^f_{FON}) = 0. \quad (37)
\]

**Proof.** Since \( \{f(X^k)\} \) is non-increasing, relationship (36) holds. Now, we assume statement (37) is not true. Then there exist \( \delta > 0 \) and a subsequence of \( \{X^k\} \), denoted as \( \{X^{k_j}\} \) such that
\[
\operatorname{dist}(X^{k_j}, \Omega^f_{FON}) \geq \delta. \quad (38)
\]
Since \( \{X^{k_j}\} \) is bounded, there exists a convergent subsequence of \( \{X^{k_j}\} \) and any accumulation point shall also satisfy the first-order optimality condition, which contradicts (38). We complete the proof. \( \square \)

## 5 Numerical Experiment

In this section, we report the numerical performance of the algorithms based on Algorithm 1. Two types of testing problems are chosen based on Examples 1.1 and 1.2. All experiments are performed in MATLAB R2016a under a Windows 10 operating system on a Dell Optiplex 9020 personal computer with an Intel® Core™ i7-4790 CPU at 3.6GHz×2 and 8GB of RAM.

### 5.1 Implementation Details

In Lemmas 3.2 and 3.3, we show that GR and GP satisfy sufficient function value reduction condition (16) if the fixed stepsize \( \tau \) is smaller than \( \rho^{-1} \). However, to obtain a good estimation of \( \rho \) is often intractable, and \( \rho^{-1} \) can be very small, which leads to slow convergence. In practice, we can use an alternating BB stepsize introduced in [7], which has been already adopted in the retraction based algorithm in [40]. More specifically, the updating rule for \( \tau \) can be described as follows
\[
\tau := \begin{cases} 
\tau_{BB1}^{\text{odd}}, & \text{for odd } k, \\
\tau_{BB2}^{\text{even}}, & \text{for even } k.
\end{cases} \quad (39)
\]
where
\[
\tau_{BB1}^{\text{odd}} := \frac{\langle J^k_{-1}, J^{k-1} \rangle}{\langle J^k_{-1}, K^{k-1} \rangle} \quad \tau_{BB2}^{\text{even}} := \frac{\langle J^{k-1}, K^{k-1} \rangle}{\langle K^k_{-1}, K^{k-1} \rangle},
\]
\[
J^{k-1} = X^k - X^{k-1}, \quad K^{k-1} = c(X^k) - c(X^{k-1}).
\]

We call GR and GP with stepsize \( \tau \) defined by (39) as GR-BB and GP-BB, respectively. In contrast, GR and GP with a fixed stepsize are called GR-F and GP-F, respectively.

CBCD will only be tested in solving quadratic problem (1.1). Therefore, in each inner iteration, the subproblem (28) restricted to the 2-dimensional subspace \( \operatorname{span} \{X_i, \nabla q_i(X_i)\} \) can be solved to the global optimality. However, in each outer iteration, the column updating order \( \{j_1, j_2, \ldots, j_p\} \) determines different types of algorithms. Usually, there are four orders:

a) **cyclic type:** \( j_i = i \), for \( i = 1, 2, \ldots, p \);

b) **random 1:** \( j_i = \lceil p \cdot \text{rand}(1, 1) \rceil \), for \( i = 1, 2, \ldots, p \) (sampling with replacement);
c) **random 2**: \( \{j_1, j_2, \ldots, j_p\} \) is a random permutation of \( \{1, 2, \ldots, p\} \) (sampling without replacement);

d) **greedy type**: for \( i = 1, 2, \ldots, p, \)

\[
    j_i := \arg \max_{j=1, \ldots, p} \left\| (I_n - W^{i-1}W^{i-1\top})\nabla f_j(X_j) \right\|_2.
\]

The corresponding CBCD are denoted as CBCD-C, CBCD-R1, CBCD-R2 and CBCD-G, respectively.

We have already shown that any iterate generated by any algorithm based on our new framework satisfies the symmetry and feasibility in (2). Hence, for the stopping criterion, we only need to check the projected gradient, \( \| (I_n - XX^\top)\nabla f(X) \|_F \). More specifically, the stopping criterion can be described as follows

\[
\| (I_n - XX^\top)\nabla f(X) \|_F < \epsilon \| \nabla f(X^0) - X^0\nabla f(X^0)^\top X^0 \|_F. \tag{40}
\]

The right-hand-side of (40) is to match the scale of the initial projected gradient. On the other hand, convergence of first-order methods may slow down as the iterates approach a stationary point, so it is critical to detect the slowdown and stop properly. It is usually beneficial to have flexible stopping rules for identifying the situation that the algorithm gets trapped in a certain region. As suggested in [40], we use the following rule based on the relative error for complement

\[
    \text{tol}_x^k := \frac{\| X^k - X^{k+1} \|_2}{\sqrt{n}} < \epsilon_x, \quad \text{mean}([\text{tol}_x^{k = \min(k,T)+1}, \ldots, \text{tol}_x^T]) < 10\epsilon_x,
\]

\[
    \text{tol}_f^k := \frac{|f(X^k) - f(X^{k+1})|}{|f(X^k)| + 1} < \epsilon_f, \quad \text{mean}([\text{tol}_f^{k = \min(k,T)+1}, \ldots, \text{tol}_f^T]) < 10\epsilon_f.
\]

We terminate the algorithm when one of the above four criteria is satisfied or a maximum iteration number MaxIter is reached. Unless otherwise specified, the default tolerance parameters are chosen as \( \epsilon = 10^{-5}, \epsilon_x = 10^{-5}, \epsilon_f = 10^{-8}, T = 5 \) and MaxIter = 3000.

### 5.2 Testing Problems

In this subsection, we introduce two types of testing problems.

The first type of testing problems is based on Example 1.2. Namely, we consider the following orthogonal constrained quadratic problems,

\[
\begin{aligned}
    \min_{X \in \mathbb{R}^{n \times p}} & \frac{1}{2} \text{tr}(X^\top AX) + \text{tr}(G^\top X) \\
\text{s.t.} & \quad X^\top X = I_p, \quad \text{tr}(X^\top AX) \tag{41}
\end{aligned}
\]

where the matrices \( A \in \mathbb{R}^{n \times n} \) and \( G \in \mathbb{R}^{n \times p} \) are randomly generated by,

\[
    A := PAP^\top, \tag{42}
\]

\[
    G := \alpha \cdot QD, \tag{43}
\]

where the matrices \( P = \text{qr} \left( \text{rand}(n,n) \right) \in \mathbb{R}^{n \times n}, Q = \text{rand}(n,p) \in \mathbb{R}^{n \times p}, Q \in \mathbb{R}^{n \times p} \) and \( Q_i = \frac{\tilde{Q}_i}{\| \tilde{Q}_i \|_2} \)
In contrast, unless specifically mentioned, the default setting of these parameters are

\[
\begin{cases}
\beta^{1-i}, \text{rand}(1,1) < \xi & \forall i = 1, 2, \ldots, n, \\
-\beta^{1-i}, \text{rand}(1,1) \geq \xi & \forall i = 1, 2, \ldots, n,
\end{cases}
\]

Here, \(n \times p\) is the variable size; parameter \(\beta \geq 1\) determines the decay of eigenvalues of \(A\); parameter \(\zeta \geq 1\) refers to the growth rate of column’s norm of \(G\). The parameter \(\alpha > 0\) represents the scale difference between the quadratic term and the linear term. When \(\alpha\) is large, the linear term dominates the objective. Parameter \(\xi \in [0, 1]\) is to determine the definiteness of \(A\). Once \(\xi = 1\), matrix \(A\) is positive definite, while \(\xi = 0\) means the negative definiteness of \(A\).

The second type of testing problems is a special case of Example 1.1. It is called Kohn-Sham total energy minimization which comes from electronic structure calculation [17]. The original Kohn-Sham equations are the Euler-Lagrange equations for the continuous total energy minimization problem. Under the planewave discretization scheme, the Kohn-Sham total energy can be transformed into a finite-dimensional approximation as the following,

\[
E_{total}(X) = \text{tr}[X^\top(\frac{1}{2} L + V_{\text{ion}})X] + \frac{1}{2} \phi(X)^\top L^\dagger \phi(X) + \phi(X)^\top \epsilon_{xc}(\phi(X)),
\]

where \(\phi(X) := \text{diag}(XX^\top)\) denotes the charge density, and \(L\) is a finite-dimensional representation of the Laplacian operator in the planewave basis. The discretized local ionic potential can be represented by a diagonal matrix \(V_{\text{ion}}\). And the matrix \(L^\dagger\) which is the discrete form of the Hartree potential corresponds to the pseudo-inverse of \(L\). The exchange correlation function \(\epsilon_{xc}\) is used to model the non-classical and quantum interaction between electrons. We are aimed to solve the following total energy minimization problem,

\[
\min_{X^\top X = I_p} E_{total}(X).
\]

It is not difficult to verify that the gradient of the energy function is \(H(X)X\), where \(H(X) = L/2 + V_{\text{ion}} + \text{Diag}(L^\dagger \phi(X)) + \text{Diag}(\mu_{xc}(\phi(X)))\) is the Kohn-Sham Hamiltonian and \(\mu_{xc}(\phi(X)) = d\epsilon_{xc}/d\phi(X)\).

### 5.3 Default Settings of Our Algorithms

In this subsection, we determine the default settings for our GR, GP and CBCD by numerical experiments.

We first compare the performance of GR-F and GP-F with different fixed stepsizes for choosing a proper value of the stepsize. The parameter \(p\) in the test is chosen as \(0.1n\), the parameter \(\zeta\) is 1.01, and the other parameters take their default values. We will compare four measurements: CPU time in seconds, total number of iterations, KKT violation and scaled function values. Since both retraction based approaches and our new framework are feasible methods, we do not report the feasibility violation \(||I^\top - X^\top X||_F||. The results with respect to the above four measurements are demonstrated in subfigures (a)-(d) of Figure 3, respectively. We choose different \(\tau\) ranging from 0.1\(\rho^{-1}\) to \(\rho^{-1}\).

From Figure 3, we observe that \(\tau = 1/3\rho\) and \(1/\rho\) are the best choices for GR-F and GP-F, respectively, in this testing problem. Hence, we choose them as stepsizes in the comparison with GR-BB and GP-BB.

Next, we perform on a set of testing problems with ten randomly generated matrices with size \(n\) ranges from 500 to 5000, and the width of variable \(p\) is still 10\%\(n\). The parameter \(\zeta\) is 1.01, and the other parameters take their default values. Numerical results of this test are illustrated in Figure 4.
From Figure 4, we notice that GR-BB and GP-BB require much fewer number of iterations and less CPU time than GR-F and GP-F, and also achieve the same first-order stationary point with comparable KKT violation. Moreover, GR-BB outperforms GP-BB in terms of CPU time and iteration number in the most cases. Thus, we choose GR-BB to represent the gradient based class of algorithms in the following comparison.

We next compare the performance among CBCD variations corresponding to different updating orders. In this comparison, we run CBCD-C, CBCD-R1, CBCD-R2 and CBCD-RG to solve the testing problems with \( n \) ranges from 1000 to 6000, \( p = 2\%n \), and all the other parameters take their default values. The numerical results are presented in Figure 5.

From Figure 5, we can see that CBCD-C, CBCD-G, and CBCD-R2 have a similar performance with respect to CPU time and iteration number, and are better than CBCD-R1. Among CBCD-C, CBCD-G, and CBCD-R2, CBCD-C performs slight better and it is easy to implement. Therefore, we will use CBCD-C to represent the CBCD class of algorithms in the following tests.
5.4 Performance Comparison on Random Problem

In this subsection, we compare the performance of our algorithms GR-BB and CBCD-C with two state-of-the-art solvers in solving a large variety of problem (41). We first choose the most robust solver OptM\(^5\) based on the algorithm in [40]. For the other existing solver for comparison, we intend to choose one from MOptQR-LS (manifold QR method with line search\(^6\) [3]), MOptQR-BB (for fair comparison, we implement the same alternating BB stepsize strategy as GR-BB to manifold QR method), and MOptTR (manifold trust-region method\(^6\) [3]). We compare MOptQR-LS, MOptQR-BB and MOptTR to solve the problem (41) with default settings. The result is illustrated in Figure 6.

![Figure 6: Performance of MOptQR with different type stepsize](image)

We can learn from Figure 6 that MOptQR-BB outperforms the other two methods in the testing problems, and hence we will choose MOptQR-BB to be the other solver to compare with our algorithms. By abuse of notation, we use MOptQR to denote MOptQR-BB hereinafter.

In the following experiments, we only compare the performance among GR-BB, CBCD-C, OptM and MOptQR. We will set the same stopping criteria as introduced in Subsection 5.1, and the tolerance takes its default value. We design six groups of testing problems, in each of which there is only one parameter varying with all the others fixed. More specifically, we describe the varying parameters of each group as the following

- Number of rows of the variable, \(n = 1000j\), for \(j = 1, 2, 3, 4, 5, 6\);
- Number of columns of the variable, \(p = 20j\), for \(j = 1, 2, 3, 4, 5, 6\);
- Decay of the eigenvalues of \(A\), \(\beta = 1.01 + 0.03j\), for \(j = 0, 1, 2, 3, 4, 5, 6, 7, 8\);
- Difference between column norms of \(G\), \(\zeta = 1.01 + 0.03j\), for \(j = 0, 1, 2, 3, 4, 5, 6, 7, 8\);
- The dominance of the linear term, \(\alpha = 10^{-2}, 10^{-1}, 1, 10, 10^{2}\);
- The definiteness of \(A\), \(\xi = 0.2(j - 1)\), for \(j = 1, 2, 3, 4, 5, 6\).

The numerical results of the above six groups of testing problems are given in Figures 7, 8, 9, 10, 11 and 12, respectively.

From the above figures, we have the following observations. All solvers reach the same function value from the same initial point. They achieve comparable KKT violation with magnitude around \(10^{-5}\). Moreover, GR-BB and CBCD-C usually have lower KKT violation than the other two in most of experiments. Among the four algorithms, CBCD-C has the lowest iteration number in all the tests, while GR-BB has the least CPU time. Except for very extreme cases, CBCD-C performs the second best in terms of CPU-time.

\(^5\)Available from http://optman.blogs.rice.edu
\(^6\)Available from http://www.manopt.org
Finally, we select all the testing problems with options in bold in the above description, and put them into performance profile experiment [9]. There are altogether $6 \times 6 \times 3 \times 3 \times 3 \times 3 = 2916$ randomly generated problems. The performance profile can eliminate the influence of a small number of difficult problems and the sensitivity of results associated with the different criteria, and also provide a way to visualize the expected performance difference among many solvers. We describe the key parameters of such test as the following. For problem $m$ and solver $s$, we denote $t_{m,s}$ to represent the CPU time or iteration number. Performance ratio is defined as $r_{m,s} := t_{m,s}/\min_s \{t_{m,s}\}$. If solver $s$ fails to solve problem $m$, the ratio $r_{m,s}$ will be set to infinity or some sufficiently large number. Finally, the overall performance of solver $s$ is defined by

$$
\pi_s(\omega) := \frac{\text{number of problems where } r_{m,s} \leq \omega}{\text{total number of problems}}.
$$

It means the percentage of testing problems can be solved in $\omega$ times of the CPU time (or iteration number) used by
Figure 10: Comparison with varying G parameter ζ

Figure 11: Comparison with varying Dominance of linear term, α

Figure 12: Comparison with varying nonnegtivity of A, ξ

the fastest algorithm by solver s. Of course, the closer πs is to 1, the better performance solver s has. The performance profile results with respect to CPU time and iteration number are given in Figure 13.

We observe that GR-BB performs best and CBCD-C performs the second best among all four algorithms in solving these 2916 testing problem in CPU time, and meanwhile CBCD-C requires the least iteration number. In addition, we also provide the average KKT violation and feasibility over these 2916 random problems in Table 2.

Table 2 shows all solvers achieve a comparable average KKT violation, feasibility and function value variance. Here, the function value variance of solver s in solving problem m can be defined as

$$\sigma_{m,s} := \frac{F_{m,s} - \min_s \{F_{m,s}\}}{1 + \min_s \{F_{m,s}\}}.$$
Performance profile of 2916 problems on CPU time and iteration number for CBCD-C, OptM, GR-BB, and MOptQR.

<table>
<thead>
<tr>
<th></th>
<th>CBCD-C</th>
<th>OptM</th>
<th>GR-BB</th>
<th>MOptQR</th>
</tr>
</thead>
<tbody>
<tr>
<td>KKT violation</td>
<td>1.6075e-05</td>
<td>2.1730e-05</td>
<td>1.9501e-05</td>
<td>2.5072e-05</td>
</tr>
<tr>
<td>Function value variance</td>
<td>6.5780e-06</td>
<td>8.1754e-06</td>
<td>3.0417e-06</td>
<td>7.9584e-06</td>
</tr>
</tbody>
</table>

Table 2: Average KKT, feasibility violation and function value

5.5 Global Property of CBCD

An interesting observation of all the experiments introduced above is that all solvers reach the same function value when they converge from randomly generated initial guess, although our problem (1) is nonconvex. Therefore we design a new experiment as the following. We construct the following problem

\[
\min_{X \in \mathbb{R}^{3 \times 2}} \frac{1}{2} \text{tr} \left( (X - X^*)^\top A (X - X^*) \right)
\]

s.t. \[X^\top X = I,\]

where \( A = \begin{bmatrix} 13/2 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \). For this special problem, we can verify that \( X^* = \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \end{bmatrix} \) is the only global minimizer, while

\[
X^I = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad X^II = \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & -1 \end{bmatrix}, \quad X^III = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}
\]

are the non-global first-order stationary points. \( X^I \) is a local minimizer, while the other two are saddle points. Then we set the initial guess from neighborhoods of the non-global stationary points, and run GR-BB, CBCD-C, OptM and MOptQR to see the different performance. More specifically,

\[
X^0 := \mathcal{P}_{S_{n,p}}(X^i + \mu \cdot \text{randn}(3, 2)), \quad \text{for } i = I, II, III
\]

\[
X^0 := \mathcal{P}_{S_{n,p}}(\text{randn}(3, 2))
\]
where \( \mu > 0 \) controls the distance between \( X^0 \) and \( X^i \) for \( i = I, II, III \). We set \( \mu = 10^{-4} \) and compare all solvers with these four types of initial points. With repeating each test 1000 times, we record the number of each solution for every solver, and the success rates are presented in Tables 3, 4, 5 and 6.

<table>
<thead>
<tr>
<th>Testing Methods</th>
<th>( X^* )</th>
<th>( X^I )</th>
<th>( X^{II} )</th>
<th>( X^{III} )</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBCD-C</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>GR-BB</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>OptM</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>MOptQR</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 3: Test results with initial point near \( X^I \)

<table>
<thead>
<tr>
<th>Testing Methods</th>
<th>( X^* )</th>
<th>( X^I )</th>
<th>( X^{II} )</th>
<th>( X^{III} )</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBCD-C</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>GR-BB</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>OptM</td>
<td>729</td>
<td>0</td>
<td>271</td>
<td>0</td>
<td>72.9%</td>
</tr>
<tr>
<td>MOptQR</td>
<td>78</td>
<td>922</td>
<td>0</td>
<td>0</td>
<td>7.8%</td>
</tr>
</tbody>
</table>

Table 4: Test results with initial point near \( X^{II} \)

<table>
<thead>
<tr>
<th>Testing Methods</th>
<th>( X^* )</th>
<th>( X^I )</th>
<th>( X^{II} )</th>
<th>( X^{III} )</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBCD-C</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>GR-BB</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>OptM</td>
<td>338</td>
<td>28</td>
<td>634</td>
<td>0</td>
<td>33.8%</td>
</tr>
<tr>
<td>MOptQR</td>
<td>5</td>
<td>5</td>
<td>990</td>
<td>0</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

Table 5: Test results with initial point near \( X^{III} \)

<table>
<thead>
<tr>
<th>Testing Methods</th>
<th>( X^* )</th>
<th>( X^I )</th>
<th>( X^{II} )</th>
<th>( X^{III} )</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBCD-C</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>GR-BB</td>
<td>656</td>
<td>344</td>
<td>0</td>
<td>0</td>
<td>65.6%</td>
</tr>
<tr>
<td>OptM</td>
<td>864</td>
<td>136</td>
<td>0</td>
<td>0</td>
<td>86.4%</td>
</tr>
<tr>
<td>MOptQR</td>
<td>774</td>
<td>226</td>
<td>0</td>
<td>0</td>
<td>77.4%</td>
</tr>
</tbody>
</table>

Table 6: Test results with random point guess

It can be observed from the above tables that the four algorithms are not necessarily convergent to same stationary points. In our tests, CBCD-C can always find the global minimizer. We are not sure whether it is a coincidence or CBCD-C has the nice property of converging to a global minimizer with great probability. The random initialization does increase chance to find global minimizer for the other three algorithms.

5.6 Kohn-Sham Total Energy Minimization

In the end of this section, we compare GR-BB with the state-of-the-art solvers in solving Kohn-Sham total energy minimization. Our test is based on the best MATLAB platform, as the best of our knowledge, for electronic structure calculation, KSSOLV [41]. KSSOLV has a friendly interface and allows researchers to investigate their own algorithms easily for different steps in electronic structure calculation. Currently, the most widely used algorithm for (47) is the self-consistent field (SCF) iteration, which is provided in KSSOLV. This is an iterative method for solving the nonlinear eigenvalue problem (KKT system of (47) briefly). Other methods focusing on discretized Kohn-Sham total energy minimization including direct constrained minimization [42] and its improved version, trust-region direct constrained minimization (TRDCM) [43] are also integrated in KSSOLV. TRDCM combines the trust-region and the subspace strategies to this special orthogonal constrained optimization problem, and its trust-region subproblem restricted to a subspace is solved by SCF. GR-BB and MOptQR are selected in this comparison as general solvers for orthogonal constrained optimization problem.

We select 18 testing problems with respect to different molecules, which are assembled in KSSOLV. Methods SCF and TRDCM running with \( \epsilon = 10^{-5} \), \( \text{MaxIter} = 200 \), and other parameters take their default values, while GR-BB and MOptQR improve their stopping criteria with \( \epsilon = 10^{-5}, \epsilon_x = 10^{-9}, \epsilon_f = 10^{-13} \), \( \text{MaxIter} = 1000 \) to get a comparable solution with other methods. It is worthy of mentioning that here the symmetry of (2) is already achieved, since the total energy function is homogeneous and hence without linear term. The stopping rule is set as \( \| (I - XX^\top) H(X) X \|_F < \epsilon \). For all of the testing algorithms, we set the same initial guess \( X^0 \) by using the function ‘getX0’, which is provided by KSSOLV. The numerical results are illustrated in Tables 7 and 8.
<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-1.5799906179e+01</td>
<td>8.68e-03</td>
<td>200</td>
<td>2509.48</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-1.5803817595e+01</td>
<td>8.15e-06</td>
<td>184</td>
<td>1595.83</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-1.5802118775e+01</td>
<td>8.42e-03</td>
<td>1000</td>
<td>2017.61</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-1.5802922328e+01</td>
<td>2.05e-03</td>
<td>1000</td>
<td>2070.80</td>
</tr>
</tbody>
</table>

ala, $n = 16879, p = 12$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-6.1161921213e+01</td>
<td>9.70e-07</td>
<td>15</td>
<td>204.20</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-6.1161921213e+01</td>
<td>5.91e-06</td>
<td>16</td>
<td>147.84</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-6.1161921213e+01</td>
<td>8.14e-06</td>
<td>65</td>
<td>142.70</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-6.1161921212e+01</td>
<td>9.78e-06</td>
<td>63</td>
<td>142.36</td>
</tr>
</tbody>
</table>

alanine, $n = 12671, p = 18$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-3.7225751363e+01</td>
<td>7.85e-07</td>
<td>12</td>
<td>85.52</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-3.7225751363e+01</td>
<td>7.33e-06</td>
<td>14</td>
<td>71.13</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-3.7225751363e+01</td>
<td>8.38e-06</td>
<td>127</td>
<td>154.06</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-3.7225751362e+01</td>
<td>9.69e-06</td>
<td>50</td>
<td>60.38</td>
</tr>
</tbody>
</table>

benzene, $n = 8407, p = 15$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-1.4420491322e+01</td>
<td>1.12e-06</td>
<td>11</td>
<td>10.09</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-1.4420491322e+01</td>
<td>5.00e-06</td>
<td>12</td>
<td>7.61</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-1.4420491322e+01</td>
<td>5.56e-06</td>
<td>49</td>
<td>8.53</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-1.4420491321e+01</td>
<td>9.84e-06</td>
<td>43</td>
<td>7.58</td>
</tr>
</tbody>
</table>

c2h6, $n = 2103, p = 7$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-8.1536091936e+01</td>
<td>1.52e-06</td>
<td>16</td>
<td>288.09</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-8.1536091937e+01</td>
<td>9.48e-06</td>
<td>15</td>
<td>171.38</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-8.1536091935e+01</td>
<td>9.51e-06</td>
<td>442</td>
<td>1296.05</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-8.1536091936e+01</td>
<td>8.85e-06</td>
<td>50</td>
<td>157.02</td>
</tr>
</tbody>
</table>

c12h26, $n = 5709, p = 37$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-3.5124395801e+01</td>
<td>1.50e-06</td>
<td>11</td>
<td>11.92</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-3.5124395801e+01</td>
<td>7.63e-06</td>
<td>13</td>
<td>8.72</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-3.5124395800e+01</td>
<td>9.03e-06</td>
<td>39</td>
<td>7.53</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-3.5124395800e+01</td>
<td>6.94e-06</td>
<td>39</td>
<td>7.52</td>
</tr>
</tbody>
</table>

co2, $n = 2103, p = 8$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-3.5124395801e+01</td>
<td>1.50e-06</td>
<td>11</td>
<td>11.92</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-3.5124395801e+01</td>
<td>7.63e-06</td>
<td>13</td>
<td>8.72</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-3.5124395800e+01</td>
<td>9.03e-06</td>
<td>39</td>
<td>7.53</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-3.5124395800e+01</td>
<td>6.94e-06</td>
<td>39</td>
<td>7.52</td>
</tr>
</tbody>
</table>

cube661, $n = 12599, p = 48$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-1.3463843176e+02</td>
<td>2.80e-06</td>
<td>13</td>
<td>532.25</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-1.3463843176e+02</td>
<td>5.77e-06</td>
<td>22</td>
<td>787.58</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-1.3463843177e+02</td>
<td>5.06e-06</td>
<td>533</td>
<td>3817.95</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-1.3463843176e+02</td>
<td>9.27e-06</td>
<td>68</td>
<td>493.53</td>
</tr>
</tbody>
</table>

glutamine, $n = 16517, p = 29$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-9.1839425243e+01</td>
<td>2.88e-06</td>
<td>17</td>
<td>616.73</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-9.1839425244e+01</td>
<td>8.49e-06</td>
<td>15</td>
<td>479.34</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-9.1839425243e+01</td>
<td>7.26e-06</td>
<td>87</td>
<td>570.86</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-9.1839425243e+01</td>
<td>9.76e-06</td>
<td>75</td>
<td>499.92</td>
</tr>
</tbody>
</table>

graphene16, $n = 3071, p = 37$

<table>
<thead>
<tr>
<th>Solver</th>
<th>$E_{tot}$</th>
<th>KKT violation</th>
<th>Iteration</th>
<th>CPU time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>-9.3873673630e+01</td>
<td>5.28e-03</td>
<td>200</td>
<td>2008.61</td>
</tr>
<tr>
<td>TRDCM</td>
<td>-9.4046217545e+01</td>
<td>6.12e-06</td>
<td>43</td>
<td>313.88</td>
</tr>
<tr>
<td>MOptQR</td>
<td>-9.4046217540e+01</td>
<td>9.56e-06</td>
<td>693</td>
<td>1110.39</td>
</tr>
<tr>
<td>GR-BB</td>
<td>-9.4046217543e+01</td>
<td>8.35e-06</td>
<td>321</td>
<td>513.45</td>
</tr>
</tbody>
</table>

Table 7: The results in total energy minimization
Table 8: The results in total energy minimization
Here, \( E_{\text{tot}} \) represents the total energy function value, and “KKT violation”, “Iteration” and “CPU time(s)” stand for \( \| (I - XX^\top)H(X)X \|_F \), the number of iteration and the total running time in second, respectively. From the tables, we observe that GR-BB outperforms other algorithms, even the heuristic ones, in most cases, and it obtains a comparable total energy function value and a lower KKT violation. In particular, in the large size problem “ctube661”, GR-BB achieves the same total energy function value and same magnitude KKT violation, but requires much less CPU time than others.

6 Conclusion

In this paper, we propose a new first-order framework, Algorithm 1, for orthogonal constrained optimization problem (1). This algorithm framework consists of two steps. In the first step, we choose any function value reduction approach in the Euclidean space to reduce the function and keep feasibility at the same time, and hence Stiefel manifold or its tangent space based calculation is not involved. In the second step, a symmetrization step is employed to guarantee that any accumulation point of the iterates is a first-order stationary point. Moreover, for some special cases, the symmetrization step can be waived. We introduce two classes of approaches. The difference of them is in the first step. We first put forward a gradient based scheme, whose global convergence can be guaranteed by fixed stepsize and hence line search is no longer needed. We recommend two particular algorithms, GR and GP, from this class. The second class of algorithm is called CBCD, in which the column-wise block coordinate update is conducted in a Gauss-Seidel manner. We also propose novel ideas to solve the column-wise subproblem efficiently and the strategy guarantees the global convergence. Preliminary experiments on two large classes of testing problems including Kohn-Sham total energy minimization arising from electronic structure calculation illustrate that our new algorithms have great potential.

However, how to design second-order methods in Euclidean space to further enhance the performance and obtain local minimizers is still under investigation. Global optimality under some random assumptions is an attractive topic for future work. How to design Jacobian type column-wise block coordinate descent method is very important for the parallelization, as low scalability is an inevitable bottleneck of existing approaches for solving orthogonal constrained problems.

Acknowledgements

The authors would like to thank Zhaosong Lu, Ting Kei Pong, Zaiwen Wen and Zaikun Zhang for the insightful discussions.

References


