Orthogonal projection algorithm for projecting onto a finitely generated cone *

Chengjin Li† and Shenggui Zhang‡
College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China

Abstract

In this paper, an algorithm is proposed to find the nearest point of a convex cone to a given vector, which is composed of a series of orthogonal projections. Some properties of this algorithm, including the reasonability of implementation, the global convergence property and the finite termination, etc., are obtained. The proposed algorithm is more stable than other related algorithms, which is verified by the numerical results shown at the end of this work.

Mathematics Subject Classifications: 65K05, 90C30
Key words. finitely generated cone, orthogonal projection, finite termination

1. Introduction

In $\mathcal{R}^m$, the finitely generated cone we are concerned about is $\{Ax|x \in \mathcal{R}^n_+\}$, where $A = (a_1, a_2, \cdots, a_n) \in \mathcal{R}^{m\times n}$ is a full-column-rank matrix and $\mathcal{R}^n_+$ denotes the non-negative part of $\mathcal{R}^n$. Thus the least squares problem related to the finitely generated cone can be formulated as

$$\min_{x\in\mathcal{R}^n_+} \quad f_A(x) := \|Ax - b\|_2^2 \tag{1.1}$$

with a given vector $b \in \mathcal{R}^m$. For simplicity, we suppose all vectors in this paper are all column vectors unless the transpose symbol $'\top'$ is added. Since Problem (1.1) can be found in a wide range of application fields, it has attracted considerable attention recently. For the more details, please refer [2, 3, 4, 5, 6, 7, 10, 11] and the reference therein.

If $m = n$, then the finitely generated cone is reducible to a simplicial cone, and the least squares problem related to the simplicial cone can be solved by many different methods, for example, Picard's method [2, 3], the semi-smooth Newton's method [4], the critical index algorithm [6] and so on. Unfortunately, these methods are not suitable for Problem (1.1) if $m \neq n$. Many iterative methods for the quadratic programming with bounds on the variables, such as the two-phase gradient method [9], can also be used to solve Problem (1.1). But, no matter what the error tolerance is, the final iteration point obtained by these iterative methods is just the approximative solution of Problem (1.1) instead of the exact optimal solution. Moreover, some gradient-type iterative methods such as the two-phase gradient method are not competent for Problem (1.1) with the ill-conditional parameter matrix A.

Some methods based on a series of orthogonal projections can be used for solving Problem (1.1), such as the subalgorithm in [11] and the algorithm in [7]. In each iteration of the subalgorithm, which is a method with the enumerative type scheme, the orthogonal projection point obtained by projecting b onto a given subspace is checked whether it is the optimal solution of Problem (1.1). For simplicity, the method in [7] is called as ORP method, and it is more efficient than the

*This work was supported by the National Natural Science Foundations of China (11301080, 11526053), Science Foundation of Fujian Province of China (2016J05003), the Foundation of the Education Department of Fujian Province of China (JA15106), and the Project of Nonlinear analysis and its applications (IRTL1206).
†E-mail: chengjin98298@163.com
‡E-mail: zsgll@fjnu.edu.cn

subalgorithm in [11] from the viewpoint of numerical experiments. But, both these two methods may not converge to the optimal solution of Problem (1.1), especially the subalgorithm in [11] would not terminate in some cases.

For ORP method, specifically speaking, there exists a defect in its implementation process: even if the optimal solution of Problem (1.1) is not the original point, there is a possibility that the final index set found by this method is empty, which together with the related assumption in [7] deduces a contradictory result, i.e., the obtained optimal solution is the original point. In order to correct this defect, an important index set is introduced to find a sequence of faces of the finitely generated cone.

In this paper, based on the new index set and the orthogonal projection, a new orthogonal projection algorithm is designed. The defect is overcome in the implementation process of the proposed algorithm. Besides, the proposed algorithm can terminate at the optimal solution of Problem (1.1) in a finite number of steps. Meanwhile, the efficiency and stability of the proposed algorithm are verified by some numerical experiments. Since Problem (1.1) with $m \neq n$ can only be solved by ORP method and the methods for the quadratic programming with bounds on the variables, comparative numerical experiments are only carried out between our algorithm and the these two methods.

The remainder of the paper is organized as follows: In Section 2, we briefly review the defect of ORP method, and introduce a new algorithm. Some related properties of the proposed algorithm are also presented in this section. The numerical results and the final conclusion are shown in Section 3 and Section 4 respectively.

2. New orthogonal projection algorithm

Let $\theta = \{1, 2, \cdots, n\}$, and for any nonempty index subset $\vartheta = \{i_1, i_2, \cdots, i_k\} \subseteq \theta$, we use A_ϑ, $cone(A_\vartheta)$, A_ϑ^+, $\Pi_\vartheta(b)$ and $P_\vartheta(b)$ to denote the submatrix $(a_{i_1}, a_{i_2}, \cdots, a_{i_k})$, the convex cone $\{\Sigma_{j=1}^k \alpha_{i_j} a_{i_j} \mid \alpha_{i_j} \geq 0 \text{ for all } j = 1, \cdots, k\}$, the matrix $(A_\vartheta^\top A_\vartheta)^{-1} A_\vartheta^\top$, the orthogonal projection point $A_\vartheta A_\vartheta^+ b$ and the projection point of b onto $cone(A_\vartheta)$, respectively. In the same time, the notations $\phi, |\vartheta|$, $\theta \setminus \vartheta$ and 0_n are used to denote respectively the empty set, the number of the elements in the set ϑ, the residual index set obtained by deleting ϑ from θ and the n-dimensional zero vector. For simplicity, we define $x \succeq 0_n (\succ 0_n) \Longleftrightarrow x_i \geq 0 (x_i > 0)$ for all $i \in \theta$, so is the notations $' \preceq '$ and $' \prec '$. At last, let $x^* = (x_1^*, x_2^*, \cdots, x_n^*)^\top$ be the optimal solution of Problem (1.1) with its active set[7] θ^* defined as

$$\theta^* = \{i \in \theta | x_i^* > 0\}. \tag{2.1}$$

If the index set θ is divided into two non-empty subsets θ_1 and θ_2, then it follows from Theorem 1 in [1] that the orthogonal projection point $\Pi_\theta(b) = AA^+ b = A_\theta A_\theta^+ b$ can be reformulated as

$$\Pi_\theta(b) = A_\theta A_\theta^+ b = A_{\theta_1} (\Pi_{\theta_2}^\perp A_{\theta_1})^+ b + A_{\theta_2} (\Pi_{\theta_1}^\perp A_{\theta_2})^+ b, \tag{2.2}$$

where $\Pi_{\theta_1}^\perp A_{\theta_2} = (\Pi_{\theta_1}^\perp (a_i))_{i \in \theta_2}$ with its column $\Pi_{\theta_1}^\perp (a_i) = a_i - \Pi_{\theta_1}(a_i)$ for each $i \in \theta_2$.

With the help of the above notations, some useful conclusions introduced in paper [7] are summarized as follows.

Lemma 2.1 [7] *(i)* $(b - \Pi_{\theta_1}(b))^\top a_i = b^\top (a_i - \Pi_{\theta_1}(a_i))$ *holds for any* $i \in \theta$;
(ii) $\{\Pi_{\theta_1}^\perp (a_i) | i \in \theta_2\}$ *is a linearly independent set, so is* $\{\Pi_{\theta_2}^\perp (a_i) | i \in \theta_1\}$;
(iii) If $a_i^\top b > 0$ *holds for all* $i \in \vartheta$, *and* $\Pi_\vartheta(b) = \Sigma_{i \in \vartheta} \alpha_i a_i$. *Then* $\alpha_i > 0$ *holds for some* $i \in \vartheta$.

A condition, which is equivalent to the first order optimality condition (KKT condition) of Problem (1.1), is shown in the following theorem.

Lemma 2.2 [7, 11] *Let* x^* *minimize* $f_A(x)$ *in (1.1), and let* θ^* *be the corresponding active set defined as that in (2.1). Then* θ^* *is uniquely characterized by the following conditions:*
(i) $A_{\theta^*}^+ b \succ 0_{|\theta^*|}$, *and*
(ii) $a_i^\top (b - \Pi_{\theta^*}(b)) \leq 0$ *for all* $i \in \theta \setminus \theta^*$.
Moreover, we have $P_\theta(b) = \Pi_{\theta^*}(b)$.

After defining the index set $\iota_+(\vartheta, b) = \{i \in \vartheta \mid$ the coefficient of a_i in the orthogonal projection $A_\vartheta A_\vartheta^+ b$ is strictly positive$\}$, the author in [7] designed an important iterative process

$$\iota_+^{k+1} = \iota_+(\iota_+^k, b) \tag{2.3}$$

with the initial index set $\iota_+^0 = \theta$. Furthermore, the iterative process (2.3) terminates if $\iota_+^{k+1} = \iota_+^k$ or $\iota_+^k = \phi$ holds for certain nonnegative integer k, and therefore this iterative process will stop after finitely many iterations. For simplicity, the final index set generated by the iterative process (2.3) is denoted by ι_+^∞. Then, in [7], the author claimed that the projection point $P_\theta(b) = \Pi_{\iota_+^\infty}(b) = \Pi_\phi(b) = 0_m$ if $\iota_+^\infty = \phi$. But this is not exact, which can be verified by the following two counterexamples.

Example 1:

$$A = \begin{pmatrix} -10 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} = (a_1, a_2), \qquad b = \begin{pmatrix} -10 \\ -1 \\ 1 \end{pmatrix}.$$

It is clear that $\theta = \{1, 2\}$ and $\Pi_\theta(b) = (-10, -1, 0)^\top = -a_1 - 20a_2$, i.e., $A^+ b = (-1, -20)^\top$ in this example. By using the iterative process (2.3), the equations $\iota_+^0 = \theta$ and $\iota_+^\infty = \iota_+^1 = \phi$ are obtained, which implies the projection point is the original point. However the real projection point $P_\theta(b)$ is $0.98a_1 = (-9.8, 0.98, 0)^\top$.

Example 2:

$$A = \begin{pmatrix} -6 & 8 & 6 \\ 2 & -1 & -1 \\ 1 & -1 & -1 \end{pmatrix} = (a_1, a_2, a_3), \qquad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Similarly, we have $\theta = \{1, 2, 3\}$, $b = \Pi_\theta(b) = -a_1 + 3a_2 - 5a_3$, $A^+ b = (-1, 3, -5)^\top$, $\iota_+^0 = \theta$, $\iota_+^1 = \{2\}$ and $\iota_+^\infty = \iota_+^2 = \phi$. But, the projection point $P_\theta(b)$ of this example is $(-0.095, 0.191, 0.048)^\top = 0.143a_1 + 0.095a_2$ instead of the original point.

In order to correct the defect, an important index set is introduced firstly:

$$\varrho = \{i \in \theta | a_i^\top b > 0\}, \tag{2.4}$$

then it is easily deduced from the statement (iii) in Lemma 2.1 that $\iota_+^\infty \neq \phi$ if the initial index set $\iota_+^0 = \varrho$ and $\varrho \neq \phi$. And another important property about ϱ is shown in the following proposition.

Proposition 2.3 *The optimal solution of Problem (1.1) is the original point if and only if $\varrho = \phi$.*

Proof. For the sufficiency, we have $A^\top b \preceq 0_n$, which implies $y^\top b \leq 0$ for all $y \in cone(A)$, and therefore the following inequality

$$\|y - b\|_2^2 = \|b\|_2^2 + \|y\|_2^2 - 2y^\top b \geq \|b\|_2^2$$

holds for all $y \in cone(A)$.

In order to prove the necessity, we assume by contradiction that there exists $i \in \theta$ such that $a_i^\top b > 0$. After setting $y_\alpha = \alpha a_i (\alpha > 0)$, we have $y_\alpha \in cone(A) \setminus \{0_m\}$ and

$$\|y_\alpha - b\|_2^2 = \|a_i\|_2^2 \alpha^2 - 2a_i^\top b\alpha + \|b\|_2^2 = \|b\|_2^2 - \alpha(2a_i^\top b - \|a_i\|_2^2 \alpha) < \|b\|_2^2$$

hold if $\alpha \in (0, (2a_i^\top b)/\|a_i\|_2^2)$, which is contradict to the optimality of the original point. □

The conditions $\varrho = \phi (\Longleftrightarrow P_\theta(b) = 0_m$ from Proposition 2.3) and $A_\theta^+ b \succeq 0_n$ ($\Longleftrightarrow P_\theta(b) = \Pi_\theta(b)$) are all extreme cases, and therefore they do not need to be discussed further. For the simplicity of the following discussion, the face of a nonempty convex set is introduced briefly as follows[8]: Let Ω and Δ be the nonempty convex sets satisfying $\Delta \subseteq \Omega$, then Δ is a face of Ω if we have $x, y \in \Delta$ under the assumptions of $x, y \in \Omega$, $\lambda \in (0, 1)$ and $\lambda x + (1 - \lambda)y \in \Delta$. For completeness, define $cone(A_\phi) = \{0_m\}$.

Lemma 2.4 *For each $\vartheta \subseteq \theta$, $cone(A_\vartheta)$ is a face of $cone(A)$.*

Proof. Clearly, $cone(A)$ and $\{0_m\}$ are both faces of $cone(A)$ on account of the full column rank of A, and they are called the trivial faces. Now, we are going to prove that $cone(A_\vartheta)$ is the non-trivial face of $cone(A)$ for each nonempty index set $\vartheta \subset \theta$.

If there exist $u, v \in cone(A)$ such that

$$\lambda u + (1-\lambda)v = w \in cone(A_\vartheta),$$

where $u = u_1 + u_2$, $v = v_1 + v_2$, $\lambda \in (0,1)$, $u_1, v_1 \in cone(A_\vartheta)$ and $u_2, v_2 \in cone(A_{\theta\setminus\vartheta})$, then

$$(\lambda u_1 + (1-\lambda)v_1 - w) + (\lambda u_2 + (1-\lambda)v_2) = 0_m. \tag{2.5}$$

Combining (2.5) and the full-column-rank of A, we see that

$$\lambda u_2 + (1-\lambda)v_2 = 0_m. \tag{2.6}$$

Let $u_2 = \Sigma_{i\in\theta\setminus\vartheta}\alpha_i a_i, v_2 = \Sigma_{i\in\theta\setminus\vartheta}\beta_i a_i$ with $\alpha_i \geq 0, \beta_i \geq 0$, then equation (2.6) can be reformulated as

$$0_m = \lambda\Sigma_{i\in\theta\setminus\vartheta}\alpha_i a_i + (1-\lambda)\Sigma_{i\in\theta\setminus\vartheta}\beta_i a_i = \Sigma_{i\in\theta\setminus\vartheta}[\lambda\alpha_i + (1-\lambda)\beta_i]a_i. \tag{2.7}$$

It follows from (2.7) and the linear dependence of $\{a_i | i \in \theta \setminus \vartheta\}$ that $\lambda\alpha_i + (1-\lambda)\beta_i = 0$ holds for all $i \in \theta \setminus \vartheta$. Therefore, for each $i \in \theta \setminus \vartheta$, we have $\alpha_i = \beta_i = 0$ from the facts that $\lambda \in (0,1)$, $\alpha_i \geq 0$ and $\beta_i \geq 0$. Hence, $v_2 = 0_m$ and $u_2 = 0_m$, which implies $u, v \in cone(A_\vartheta)$. □

In this work, a face $cone(A_\vartheta)$ is called as the positive face related to the given vector b if $A_\vartheta^+ b \succ 0_{|\vartheta|}$, from which one deduces that $\Pi_\vartheta(b) = P_\vartheta(b)$ if $cone(A_\vartheta)$ is a positive face. Obviously, the optimal face $cone(A_{\theta^*})$ is the positive face containing the optimal solution of Problem (1.1), and the purpose of the iterative process (2.3) is to find the positive face related to b by using several orthogonal projections. Let

$$\theta_- = \{i \in \theta | (A^+ b)_i \leq 0\},$$

where $(A^+b)_i$ denotes the i-th component of $A^+b \in \mathcal{R}^n$. Then some useful conclusions related to θ_- and ϱ are introduced before the proposed algorithm is designed.

Proposition 2.5 *(i). If $\theta_- \neq \phi$, then there exists at least one index $i \in \theta_-$ such that $a_i \notin cone(A_{\theta^*})$;*

(ii). If $\varrho = \{j\}$, then $a_j \in cone(A_{\theta^*})$;

(iii). For each a_i with $i \in \theta$, the coefficient of a_i in the expression of $\Pi_\theta(b)$ and the value $(b - \Pi_{\theta\setminus\{i\}}(b))^\top a_i$ have the same sign (including positive, negative and zero).

Proof. If $\theta_- = \theta$, then the statement (i) holds directly from Lemma 2.2. Otherwise, $\theta_- \subset \theta$. In this case, it is easy to verify from the statement (i) in Lemma 2.2 that $\theta \neq \theta^*$, which implies the only thing left is to prove $\tilde{\theta} \neq \theta^*$ holds for each index set $\tilde{\theta}$ satisfying $\theta_- \subseteq \tilde{\theta} \subset \theta$. It follows from (2.2) that

$$\Pi_\theta(b) = A_{\tilde{\theta}}(\Pi_{\theta\setminus\tilde{\theta}}^\perp A_{\tilde{\theta}})^+ b + A_{\theta\setminus\tilde{\theta}}(\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^+ b. \tag{2.8}$$

Then, combining (2.8) and the fact $\theta_- \subseteq \tilde{\theta}$, we see that

$$(\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^+ b \succ 0_{|\theta\setminus\tilde{\theta}|}. \tag{2.9}$$

Let $Q = ((\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^\top (\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}}))^{-1}$ and $d = (a_{k_1} - \Pi_{\tilde{\theta}}(a_{k_1}), \cdots, a_{k_{|\theta\setminus\tilde{\theta}|}} - \Pi_{\tilde{\theta}}(a_{k_{|\theta\setminus\tilde{\theta}|}}))^\top b$, where $k_j \in \theta \setminus \tilde{\theta}$ for $j = 1, \cdots, |\theta \setminus \tilde{\theta}|$. Then, according to the definitions of A_ϑ^+ and $\Pi_{\theta_1}^\perp A_{\theta_2}$, it follows that

$$(\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^+ b = ((\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^\top (\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}}))^{-1} (\Pi_{\tilde{\theta}}^\perp A_{\theta\setminus\tilde{\theta}})^\top b = Qd. \tag{2.10}$$

Furthermore, the statement (ii) in Lemma 2.1 implies the positive definiteness of Q, which together with (2.9) and (2.10) deduces that $d \neq 0_{|\theta\setminus\tilde{\theta}|}$. Hence, it turns out that

$$d^\top Q d > 0. \tag{2.11}$$

We assume by contradiction that $cone(A_{\tilde{\theta}})$ is the optimal face of the original problem, then

$$d = ((b - \Pi_{\tilde{\theta}}(b))^\top A_{\theta\setminus\tilde{\theta}})^\top \preceq 0_{|\theta\setminus\tilde{\theta}|}, \tag{2.12}$$

where the equation comes from the statement (i) in Lemma 2.1 and the inequality comes from the statement (ii) in Lemma 2.2. From (2.9) and (2.12) one deduces that

$$d^\top Q d = d^\top (\Pi_{\tilde{\theta}}^{\perp} A_{\theta\setminus\tilde{\theta}})^+ b \leq 0, \tag{2.13}$$

which is contradict to (2.11).

For (ii), it can be easily verified from Proposition 2.3 that the projection point by projecting b onto each face without $\{a_j\}$ is the original point 0_m. Thus, we know the faces without $\{a_j\}$ would not be the optimal faces by considering the facts that $0_m \neq P_{\{j\}}(b) = \Pi_{\{j\}}(b) \in cone(A)$ and $\|b - P_{\{j\}}(b)\|_2^2 < \|b\|_2^2$ obtained from the definition of ϱ.

For (iii), from (2.2) one deduces that

$$\Pi_\theta(b) = a_i(\Pi_{\theta\setminus\{i\}}^{\perp}(a_i))^+ b + A_{\theta\setminus\{i\}}(\Pi_{\{i\}}^{\perp} A_{\theta\setminus\{i\}})^+ b,$$

and therefore, the coefficient of a_i is $(\Pi_{\theta\setminus\{i\}}^{\perp}(a_i))^+ b = \|a_i - \Pi_{\theta\setminus\{i\}}(a_i)\|_2^{-2}(a_i - \Pi_{\theta\setminus\{i\}}(a_i))^\top b = \|a_i - \Pi_{\theta\setminus\{i\}}(a_i)\|_2^{-2}(b - \Pi_{\theta\setminus\{i\}}(b))^\top a_i$, where the equations come respectively from the statements (ii) and (i) in Lemma 2.1. The proof is completed. □

The statement (i) in Proposition 2.5 has been mentioned in [11], but we prove it in a different way. Lemma 2.2 claims that the optimal solution can be determined if the corresponding active set θ^* is found, and therefore, the proposed algorithm terminates if θ^* is obtained. Then, based on the above discussion, the new orthogonal projection algorithm is designed as follows.

Algorithm 2.6 *New orthogonal projection algorithm for Problem (1.1)*

(S.0) Input $A \in \mathcal{R}^{m\times n}$ with rank n and $b \in \mathcal{R}^m$, and compute ϱ as that in (2.4). If $\varrho = \phi$, stop and the optimal solution is the original point 0_m; Otherwise go to Step (S.1).

(S.1) Compute $A^+ b$. If $A^+ b \succeq 0_n$, stop and the optimal solution is the orthogonal projection point $\Pi_\theta(b)$; Otherwise, find the nonempty index set ι_+^∞ by using (2.3) with $\iota_+^0 = \varrho$. Let $\xi = \iota_+^\infty$ and

$$\bar{\theta} = \begin{cases} \theta_- \setminus \varrho, & \text{if } |\varrho| = 1 \text{ and } \varrho \cap \theta_- \neq \phi, \\ \theta_-, & \text{otherwise}, \end{cases} \tag{2.14}$$

go to Step (S.2).

(S.2) Set $\rho_1 = \{i \in \theta \setminus \xi | a_i^\top (b - \Pi_\xi(b)) > 0\}$ and $\rho_2 = (\theta \setminus \xi) \cap \theta_-$. If $\rho_1 = \phi$, then $\theta^* = \xi$ with the corresponding optimal solution being $\Pi_\xi(b)$, and stop; Otherwise go to (S.3).

(S.3) Determine the index set ϑ by

$$\vartheta = \begin{cases} \xi \cup \rho_1, & \text{if } |\rho_1| < |\theta \setminus \xi|, \\ \theta \setminus \{j\} \text{ with } j \in \rho_2, & \text{if } |\rho_1| = |\theta \setminus \xi| \geq 2 \text{ and } |\rho_2| > 0, \\ \theta \setminus \{j\} \text{ with } j \in \theta \setminus \xi, & \text{if } |\rho_1| = |\theta \setminus \xi| \geq 2 \text{ and } \rho_2 = \phi, \\ \theta \setminus \theta_-, & \text{if } |\rho_1| = |\theta \setminus \xi| = 1 \text{ and } |\theta_-| = 1, \\ \theta \setminus \{j\} \text{ with } j \in \bar{\theta}, & \text{if } |\rho_1| = |\theta \setminus \xi| = 1 \text{ and } |\theta_-| > 1 \text{ (Restart phase)}, \end{cases} \tag{2.15}$$

and set $\bar{\theta} = \bar{\theta} \setminus \{j\}$ if Restart phase—the last case in (2.15) happens. Construct the index subset $\eta \subset \theta$ by collecting the subscript of a_i $(i \in \theta)$ which belongs to the optimal face obtained by using Algorithm 2.6 to sovle the subproblem $\min_{x \in \mathcal{R}_+^{|\vartheta|}} f_{A_\vartheta}(x)$. Set $\xi = \eta$ and go to (S.2).

From Proposition 2.3 and the definition of the positive face, we know that the optimal solution of Problem (1.1) in the extreme case is found by Algorithm 2.6 if one of the stopping conditions in $(S.0)$ or $(S.1)$ is satisfied. In the following discussion, we suppose Algorithm 2.6 does not terminate before Step $(S.2)$, that is to say the stopping conditions in $(S.0), (S.1)$ do not hold, which together with what was discussed after (2.4) deduces that

$$\theta_- \neq \phi, \qquad \varrho \neq \phi, \qquad \theta \supset \iota_+^\infty \neq \phi. \tag{2.16}$$

To obtain the implementing reasonability of Algorithm 2.6, several related conclusions are needed.

Lemma 2.7 *The definition of $\bar{\theta}$ in (2.14) is reasonable.*

Proof. The occurrence of $\bar{\theta}$ implies that the stopping conditions in $(S.0), (S.1)$ do not hold, and therefore $\theta_- \neq \phi$ comes from (2.16). Hence, our main purpose is to show that $|\theta_-| \neq 1$ if $|\varrho| = 1$ and $\varrho \cap \theta_- \neq \phi$.

We assume by contradiction that $|\theta_-| = 1$, which yields $\{j\} = \varrho = \theta_-$. Thus, on one hand, $\{j\} = \varrho$, then we have $a_j \in cone(A_{\theta^*})$ by considering the statement (ii) in Proposition 2.5. On the other hand, $\{j\} = \theta_-$, which together with the statement (i) in Proposition 2.5 deduces that $a_j \notin cone(A_{\theta^*})$. It is a contradiction, and therefore $|\theta_-| > 1$. $\square$

It is easy to see that the index set ξ, which is related to the positive face, is updated constantly in the loop of $(S.2) \Longleftrightarrow (S.3)$ if the stopping conditions in Algorithm 2.6 do not hold.

Lemma 2.8 *In the loop of $(S.2) \Longleftrightarrow (S.3)$, the value of $\|P_\xi(b) - b\|_2^2 = \|\Pi_\xi(b) - b\|_2^2$ is strictly decreasing with the changing of ξ if one of the first four cases in (2.15) happens.*

Proof. Obviously, $cone(A_\xi)$ is a positive face for each index set ξ in Algorithm 2.6, that is to say $(A_\xi)^+ b \succ 0_{|\xi|}$, which implies that $P_\xi(b) = \Pi_\xi(b)$.

From the definition of ρ_1, we know there exists $i \in \bar{\xi} \setminus \xi$ such that $a_i^\top (b - \Pi_\xi(b)) > 0$ holds for the first three cases in (2.15), where $\bar{\xi} = \rho_1$ for the first case and $\bar{\xi} = \theta \setminus \{j\}$ for the second and third cases. Similar to the proof of Proposition 1 in [7], it is suffices to show that $P_{\xi \cup \bar{\xi}}(b) \neq P_\xi(b)$ because the projection point onto a convex set is unique. Since $A_\xi^+ b \succ 0_{|\xi|}$ yields $P_\xi(b) = \Pi_\xi(b)$, it remains to show that $P_{\xi \cup \bar{\xi}}(b) \neq \Pi_\xi(b)$. Lemma 2.2, together with (2.1) and the fact of $cone(A_\xi)$ being a positive face, deduces that $P_{\xi \cup \bar{\xi}}(b) = \Pi_\xi(b)$ if and only if $\xi = (\xi \cup \bar{\xi})^*$, where the definition of the active set $(\xi \cup \bar{\xi})^*$ is similar to that in (2.1). But since $a_i^\top (b - \Pi_\xi(b)) > 0$ holds for certain $i \in \bar{\xi} \setminus \xi$ mentioned above, it follows from Lemma 2.2 that $\xi \neq (\xi \cup \bar{\xi})^*$, and this implies $P_{\xi \cup \bar{\xi}}(b) \neq \Pi_\xi(b)$ as required.

For the forth case in (2.15), we know from the statement (i) in Proposition 2.5 that the optimal face $cone(A_\eta)$ of the subproblem is exactly the optimal face $cone(A_{\theta^*})$ of the original problem, which implies that the functional value $\|P_\xi(b) - b\|_2^2$ is decreasing. $\square$

Remark 2.9 *For the second and forth cases in (2.15), the inequality $(b - \Pi_{\theta \setminus \{j\}}(b))^\top a_j \leq 0$ comes from the fact of $j \in \theta_-$ and the statement (iii) in Proposition 2.5. Thus, if $cone(A_{\theta \setminus \{j\}})$ with $j \in \theta_-$ is a positive face, then it follows from Lemma 2.2 that $\theta \setminus \{j\} = \theta^*$.*

Lemma 2.10 *$\phi \subset \xi \subset \theta$ holds for each index set ξ in Algorithm 2.6.*

Proof. Similar to that in Lemma 2.7, the occurrence of ξ yields (2.16), which together with the statement (iii) in Lemma 2.1 deduces $\phi \subset \xi \subset \theta$ if $\xi = \iota_+^\infty$.

Otherwise, $\xi = \eta$ in the loop $(S.2) \Longleftrightarrow (S.3)$, then the relation $\xi \subset \theta$ comes from the definition of η and the number of columns of the parameter matrix in the subproblem occurring in Step $(S.3)$. Thus, it remains to show that $\eta \neq \phi$ in the following.

From Proposition 2.3, we have $\eta = \phi$ if and only if the optimal face obtained in $(S.3)$ is $\{0_m\}$, and the optimal value in this case is $\|b\|_2^2$. Hence, this proof is completed if each objective functional value $\|P_\eta(b) - b\|_2^2$ of the subproblem in $(S.3)$ is verified less than $\|b\|_2^2$. For the nonempty initial

setting of ξ in $(S.1)$—ι_+^∞, the corresponding objective functional value $\|P_\xi(b)-b\|_2^2$ is less than $\|b\|_2^2$ from the fact that $\iota_+^\infty \subseteq \varrho$.

For the first four cases of (2.15) in the loop $(S.2) \Longleftrightarrow (S.3)$, Lemma 2.8 implies that the optimal value $\|P_\eta(b)-b\|_2^2$ is less than $\|P_{\iota_+^\infty}(b)-b\|_2^2$, and therefore $\xi = \eta \neq \phi$ in these cases. In the loop $(S.2) \Longleftrightarrow$ Restart phase, by considering the definition of $\bar{\theta}$ in (2.14), we know

$$(\theta \setminus \{j\}) \cap \varrho \neq \phi \tag{2.17}$$

holds for all $j \in \bar{\theta}$. Combining (2.17) and Proposition 2.3, we see that the optimal value of the subproblem in this case is less than $\|b\|_2^2$. $\square$

From Lemma 2.10, one deduce that $\xi \neq \phi$ and $\theta \setminus \xi \neq \phi$ hold in the implementation process of Algorithm 2.6, which implies the definition of ρ_1 is reasonable.

Remark 2.11 *From Lemma 2.10 and the definition of ξ in Algorithm 2.6, we have $\phi \subset \xi \subset \theta$ and $A_\xi^+ b \succ 0_{|\xi|}$ hold for each index set ξ, which together with Proposition 2.5 and the definition of positive face deduces that $cone(A_\xi)$ is a non-trivial positive face of $cone(A)$ related to b. Hence, the implementation process in the loop $(S.2) \Longleftrightarrow (S.3)$ of Algorithm 2.6 is to push the iteration point $P_\xi(b) = \Pi_\xi(b)$ from a non-trivial positive face to another one.*

The reasonability of the implementation process of Algorithm 2.6 is guaranteed by Lemma 2.7 and Lemma 2.10. We are now in a position to deduce the finite termination of Algorithm 2.6.

Theorem 2.12 *Algorithm 2.6 will stop at the optimal solution of Problem (1.1) after finitely many iterations.*

Proof. It is easy to see from (2.15) that the number of columns of the parameter matrix in the subproblem

$$\min_{x \in \mathcal{R}_+^{|\vartheta|}} \ f_{A_\vartheta}(x) \tag{2.18}$$

in $(S.3)$ is strictly smaller than that of Problem (1.1), i.e., $|\vartheta| < n$. Therefore, as long as the step $(S.3)$ is preformed, the original problem solving is achieved by solving a series of lower dimensional problems (2.18). As a result, the optimal solution of Problem (1.1) can be found by Algorithm 2.6 after finite iterations if Problem (1.1) with $n = 1$ can be solved correctly by Algorithm 2.6 and all the positive face $cone(A_\xi)$ do not occur infinitely during the iterative procession in the loop $(S.2) \Longleftrightarrow (S.3)$.

When $n = 1$, let $A = a \in \mathcal{R}^m$. Then the corresponding optimal solution is the original point if $a^\top b \leq 0$, otherwise the optimal solution is $P_A(b) = \Pi_A(b) = a^\top b \|a\|_2^{-2} a$. The optimal solution in both cases can be easily obtained by Algorithm 2.6. So, in the following discussion, we suppose $n > 1$ and all the stopping conditions in Algorithm 2.6 do not hold.

Under the assumptions mentioned above, the index set ξ is updated constantly in the loop $(S.2) \Longleftrightarrow (S.3)$. It is easy to see from Lemma 2.8 that, once the elements in the index set ξ is changed, the value $\|P_\xi(b)-b\|_2^2$ is strictly decreasing if one of the first four cases in (2.15) happens. Therefore, in the loop $(S.2) \Longleftrightarrow (S.3)$, the positive face $cone(A_\xi)$ would not occur repeatedly except the conditions in Restart phase is satisfied. At the same while, the occurrence number of Restart phase is not more than $|\bar{\theta}|$ from the update of $\bar{\theta}$ in $(S.3)$. Hence, all the positive faces $cone(A_\xi)$ iterate finitely often, which together with the finiteness of the positive face guarantees the loop terminates after finitely many iterations.

At last, if Restart phase occurs repeatedly, then by considering (2.14) and the statement (i) and (ii) in Proposition 2.5, we know that there exists at least one index $i \in \bar{\theta}$ such that $a_i \notin cone(A_{\theta^*})$ in Restart phase, which guarantee that the optimal solution will be found finally by solving certain subproblem $\min_{x \in \mathcal{R}_+^{|\theta \setminus \{j\}|}} \ f_{A_{\theta \setminus \{j\}}}(x)$ in the process of deleting $j \in \bar{\theta}$ in turn. Hence, the optimal solution can be found before $\bar{\theta}$ in (2.15) becoming empty. $\square$

Besides finite termination, another important property of Algorithm 2.6 is its stability. For example, the parameters in Problem (1.1) are generated in Matlab:

$$A = randn(100, 100);\quad A(:, 2) = 0.9999999999 * A(:, 1) + 0.0000000001 * A(:, 2);\quad b = randn(100, 1);$$

then the smallest singular value of A belongs to the interval $(5 * 10^{-13}, 10^{-10})$ with high possibility, and this kind of problem can be solved by Algorithm 2.6.

3. Numerical Results

All the algorithms are performed on Matlab R2013b, and the corresponding numerical results reported later are obtained from a PC with 3.46G memory, Intel(R) Core(TM) i5-4590 3.30GHz CPU and win32-bit Windows 7.

Due to the lack of the resource of Matlab's code, the programs of ORP method and the sub-algorithm in [11] are also written by the authors of this work. And the deleting order of index i_k introduced ambiguously in [7] is set in this paper as follows: the elements in the index sets ρ_1 and $\theta \setminus \rho_1$ are deleted in turn.

Firstly, the numerical results by comparing ORP method with Algorithm 2.6 are shown in the following tables. In each experiment, the parameters are generated randomly: $A = randn(m, n)$, $b = randn(m, 1)$. Moreover, for fixed dimensions m, n, Problem 1.1 with k_{total} different kinds of parameters A and b are solved respectively by ORP method and Algorithm 2.6, and only the average number of the k_{total} running times of these algorithms for Problem (1.1) are shown in the following tables. At last, the notations $'time'$, $'k'_{fail}$ in the following tables denote respectively the spending time (unit: second), the number of the times ORP method can not solve Problem (1.1) correctly.

Table 1: Comparison between ORP methods and Algorithm 2.6 for Problem (1.1) with $m = n$

Parameter	ORP method		Algorithm 2.6
(n, k_{total})	k_{fail}	$time$	$time$
(3,10000)	1672	6.31e-5	6.91e-5
(5,10000)	1544	1.09e-4	1.15e-4
(8,10000)	905	1.69e-4	1.83e-4
(10,10000)	603	2.23e-4	2.34e-4
(20,10000)	70	6.87e-4	8.52e-4
(50,1000000)	11	0.0040	0.0041
(80,30000000)	1	0.0123	0.0130
(100,500)	0	0.0217	0.0228
(200,500)	0	0.1962	0.2063
(300,500)	0	0.8407	0.9234
(500,100)	0	8.4720	8.7751
(1000,100)	0	288.10	317.12

Table 2: Comparison between ORP methods and Algorithm 2.6 for Problem (1.1) with $m \neq n$

Parameter	ORP method		Algorithm 2.6
(m, n, k_{total})	k_{fail}	$time$	$time$
(5,3,10000)	968	5.97e-5	6.44e-5
(8,5,10000)	693	8.75e-5	9.05e-5
(20,8,10000)	157	1.21e-4	1.29e-4
(20,10,10000)	75	1.55e-4	1.66e-4
(60,20,50000)	2	3.51e-4	4.71e-4

Parameter	ORP method		Algorithm 2.6
(m, n, k_{total})	k_{fail}	$time$	$time$
(160,50,500)	0	0.0015	0.0017
(200,80,500)	0	0.0042	0.0052
(300,100,500)	0	0.0075	0.0095
(500,200,500)	0	0.0513	0.0828
(600,300,500)	0	0.2529	0.4261
(1000,500,100)	0	1.7366	2.8973
(1600,1000,100)	0	44.01	56.94

On one hand, by considering the data in Table 1 and the numerical results in [7], we see that Algorithm 2.6 is more efficient than the algorithm introduced in [3]. On the other hand, although the running time spent by Algorithm 2.6 is longer than that of ORP method a little bit for the same problem, the proposed algorithm is more stable than Algorithm 2.6 because there exist nonzero data in the k_{fail} columns of the above tables. Hence, corresponding to the convergence theorem of ORP method in [7], Theorem 2.12 is reasonable.

Secondly, we are going to show the numerical results of comparing the subalgorithm in [11] with Algorithm 2.6 for Problem (1.1) with $m = n$.

Table 3: Comparison between the subalgorithm in [11] and Algorithm 2.6

Parameter	the subalgorithm in [11]		Algorithm 2.6
(n, k_{total})	k_{fail}	$time$	$time$
(5,300000)	26061	1.60e-4	1.18e-4
(8,300000)	23021	9.87e-4	1.92e-4
(10,300000)	23467	0.0052	2.73e-4
(12,10000)	822	0.043	3.33e-4
(15,10000)	897	1.747	4.68e-4
(18,5000)	403	93.5	6.59e-4
(20,1000)	150	592.6	0.0013
(30,100)	88	9905	0.0024

As indicated in Table 3, we know Algorithm 2.6 is more efficient and stable than the subalgorithm in [11]. The notation k_{fail} denotes the number of the times the subalgorithm in [11] fail to solve Problem (1.1), and the failure of the subalgorithm in [11] means that the number of iterations increase to 1e9 before it find the exactly solution.

At last, our task is to show the numerical result of comparing the two-phase gradient method (abbreviated as TPG method) in [9] and Algorithm 2.6 for solving Problem (1.1). The TPG method is an iterative-type algorithm, which implies that the solution obtained by TPG method is an approximative solution of Problem (1.1). In the options—the last inputting term of TPG method, all the maximum numbers of the different iterations are set large enough. Since TPG method is designed for quadratic programming, the objective function in Problem (1.1) must be reformulated as

$$\frac{1}{2}x^\top(2A^\top A)x - (2A^\top b)^\top x + \|b\|_2^2,$$

and the inputting terms H, c and the initial iteration point of TPG method are $2A^\top A$, $2A^\top b$ and the original point, respectively.

In the following table, $k_{total} = 5000$, and the three cases are separated according to the different setting of the abstract error($AbsTol$) and the relative error($RelTol$). Specifically, $Case\ 1 : AbsTol = 1e-6$, $RelTol = 1e-6$; $Case\ 2 : AbsTol = 1e-8$, $RelTol = 8e-10$; $Case\ 3 : AbsTol = 1e-10$, $RelTol = 4e-12$. By $'val'$, we denote the average number of k_{total} values $\|x^* - xk\|_2$, where x^* and xk is the optimal solution obtained by Algorithm 2.6 and TPG method respectively.

Table 4: Comparison between TPG methods and Algorithm 2.6 for Problem (1.1)

Parameter	TPG method						Algorithm 2.6
	Case 1		*Case* 2		*Case* 3		
(m, n)	*val*	*time*	*val*	*time*	*val*	*time*	*time*
(16, 16)	1.313e-4	0.0037	1.312e-4	0.0040	1.312e-4	0.0040	5.831e-4
(20, 20)	1.584e-7	0.0048	6.196e-12	0.0054	2.850e-14	0.0056	8.132e-4
(30, 30)	4.067e-7	0.0052	4.624e-11	0.0069	5.663e-13	0.0102	0.0099
(50, 50)	8.497e-7	0.0058	3.072e-10	0.0065	8.827e-13	0.0198	0.0148
(80, 80)	1.119e-6	0.0061	6.413e-10	0.0072	4.283e-12	0.0980	0.0197
(100, 100)	1.240e-6	0.0064	7.888e-10	0.0077	4.932e-12	0.0630	0.0315
(30, 16)	9.385e-8	0.0024	8.886e-12	0.0029	7.185e-15	0.0030	3.648e-4
(80, 20)	9.801e-8	0.0025	4.173e-11	0.0028	1.149e-13	0.0064	0.0040
(80, 30)	1.828e-7	0.0029	8.771e-11	0.0033	7.138e-12	0.0102	0.0076
(160, 50)	1.853e-7	0.0029	1.306e-10	0.0034	1.165e-13	0.0081	0.0079
(160, 80)	3.761e-7	0.0039	2.591e-10	0.0047	2.014e-12	0.0105	0.0149
(180, 100)	4.434e-7	0.0055	3.246e-10	0.0063	2.194e-12	0.0106	0.0211

It is observed from Table 3 that the smaller values of *AbsTol* and *RelTol* are set, the more approximative solution is obtained. But, meanwhile, the higher precision of the solution obtained by TPG method implies the more running time it spends.

The 100×100 ill-conditional parameter matrix A is constructed in the following experiment. To be specific, the parameters are generated as follows: $A = randn(100, 100)$; $[U, D, V] = svd(A)$;

$$
\begin{aligned}
&for \;\; i = 1 : 6 \\
&\quad D((i-1) * 5 + 1, (i-1) * 5 + 1) = \alpha * D((i-1) * 5 + 1, (i-1) * 5 + 1); \\
&end
\end{aligned}
$$

$A = U * D * V'$; $b = randn(100, 1)$; with the weight parameter $\alpha \in [1e3, 1e5]$. Since the code linesearch2.m in TPG method sometimes can not work correctly, TPG method possibly terminate without solution, and the feedback of linesearch2 in this case is 'The problem is unbounded from below'. Similar to Table 1 and Table 2, the notation k_{fail} denotes the number of the times that TPG method can not work correctly. In this experiment, we set $AbsTol = 1e-7$, $RelTol = 1e-10$ and $k_{total} = 500$.

Table 5: Comparison between TPG methods and Algorithm 2.6 for Problem (1.1) with A ill-conditional

Parameter	TPG method		Algorithm 2.6
α	k_{fail}	*time*	*time*
3e3	118	9.895	9.173
4e3	214	10.15	9.269
5e3	310	10.41	8.886
6e3	395	10.04	9.241
8e3	454	9.685	9.124
1e4	474	8.488	9.180

From the data shown in Table 4, we know Algorithm 2.6 is more stable than TPG method for solving Problem (1.1) with the ill-conditional parameter matrix A.

4. Conclusion

An alternative strategy aims to solve Problem (1.1) with high efficiency and stability is constructed by using ORP method together with Algorithm 2.6. The proposed algorithm is more

efficient than the algorithm introduced in [3] for the least squares problem related to the simplicity cone, and it is still a competent method compared with TPG method in [9]. In the aspect of efficiency, the proposed algorithm is not better than ORP method in [7], but the proposed algorithm has an important quality that ORP method do not have—the reasonability of algorithmic implementation.

References

[1] Baksalary J. K. and Baksalary O. M., Particular formulae for the Moore-Penrose Inverse of a columnwise partitioned matrix, Linear Algebra and its Application, 421(2007), 16-23.

[2] Barrios J., Ferreira O. P. and Nemeth S. Z., Projection onto simplicial cones by Picard's method, Linear Algebra and its Application, 480(2015), 27-43.

[3] Ekart A., Nemeth A. B. and Nemeth S. Z., Rapid heuristic projection on simplicial cones, (2010), ArXiv e-prints.

[4] Ferreira O. P. and Nemeth S. Z., Projection onto simplicial cones by semi-smooth Newton's method, Optimization Letter, 9(2015), 731-741.

[5] Hu X., An exact algorithm for projection onto a polyhedral cone, Australian and New Zealand Journal of Statistics, 40:2(1998), 165-170.

[6] Murty K. and Fathi Y., A critical index algorithm for nearest point problems on simplicial cones, Mathematical Programming, 23(1982), 206-215.

[7] Oh K. K., Algorithm for projecting onto simplicial cones and application to portfolio optimization, preprint, 2017.

[8] Rockafellar. R. T., Convex analysis, Princeton Unversity Press, 1970.

[9] Serafino D. di, Toraldo G., Viola M. and Barlow J., A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables, SIAM Journal on Optimization, 28(2018), 2809-2838.

[10] Ujvari M., On the projection onto a finitely generated cone, Acta Cybernetica, 22(2016), 657-672.

[11] Zheng Y. and Chew C. M., Distance between a point and a convex cone in n-dimension space: computation and applications, IEEE Transactions on Robotics, 25(2009), 1397-1412.