

Research Article

Preconditioned ADMM for a Class of Bilinear Programming Problems

Xiaobo Liang¹ and Jianchao Bai² 

¹Department of Mathematics and Computer Science, Guilin Normal College, Guilin 541001, China

²School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

Correspondence should be addressed to Jianchao Bai; bjc1987@163.com

Received 29 April 2017; Revised 19 August 2017; Accepted 21 August 2017; Published 10 January 2018

Academic Editor: S. S. Ravindran

Copyright © 2018 Xiaobo Liang and Jianchao Bai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We design a novel preconditioned alternating direction method for solving a class of bilinear programming problems, where each subproblem is solved by adding a positive-definite regularization term with a proximal parameter. By the aid of the variational inequality, the global convergence of the proposed method is analyzed and a worst-case $\mathcal{O}(1/t)$ convergence rate in an ergodic sense is established. Several preliminary numerical examples, including the Markowitz portfolio optimization problem, are also tested to verify the performance of the proposed method.

1. Introduction

Let \mathcal{R} , \mathcal{R}^n , $\mathcal{R}^{m \times n}$ be the set of real numbers, the set of n -dimensional real column vectors, and the set of $m \times n$ real matrices, respectively. For any $x, y \in \mathcal{R}^n$, we use the symbol $\langle x, y \rangle = x^T y$ to denote their inner product and the symbol $\|x\| = \sqrt{\langle x, x \rangle}$ to stand for the Euclidean norm of x , where the superscript T is the transpose. Symbol I_n is $n \times n$ identity matrix, which is simply denoted by I with proper dimension in the context. Consider the following generalized bilinear programming:

$$\min \{f(x) \mid A_1 x = b_1, A_2 x = b_2, x \in \mathcal{X} \cap \mathcal{Y}\}, \quad (1)$$

where $f(x) : \mathcal{R}^n \rightarrow \mathcal{R}$ is a closed proper convex function (possibly nonsmooth); $A_i \in \mathcal{R}^{m_i \times n}$, $b_i \in \mathcal{R}^{m_i}$ ($i = 1, 2$) are given matrices and vectors, respectively; \mathcal{X} and \mathcal{Y} are closed convex sets with nonempty intersection. Throughout this article, we assume that the solution set of (1) is nonempty.

The bilinear programming (1) arises in many applications, for instance, the Linear-Max-Min problem, the Location-Allocation problem, and the classic Markowitz portfolio optimization problem; see, for example, [1, 2] for more details.

As an extension of the linear programming, problem (1) can be recast as the following existing model:

$$\min \{f(x) \mid Ax = b, x \in \Omega\} \quad (2)$$

with

$$\begin{aligned} A &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \in \mathcal{R}^{(m_1+m_2) \times n}, \\ b &= \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in \mathcal{R}^{m_1+m_2}, \\ \Omega &= \mathcal{X} \cap \mathcal{Y}, \end{aligned} \quad (3)$$

which can be solved by some classical optimization methods if the constrained set Ω is easy, such as the proximal point algorithm [3] and the augmented Lagrangian method [4]. However, using such transformations will lead to a large-scale coefficient matrix and increase the computational complexity and storage requirement. In particular, when sets \mathcal{X} and \mathcal{Y} are complex or the objective function is not well defined, we would better deal with the variables separately and make the best of the structure properties of the given sets.

Next, we focus on developing a preconditioned alternating direction method of multipliers (P-ADMM) for solving (1), where each subproblem can be solved by adding a proximal regularization term and the Lagrange multipliers can be updated by using some preconditioned symmetric positive-definite (SPD) matrices. By the aid of new variables x_1 and x_2 , problem (1) is firstly reformulated as follows:

$$\begin{aligned} \min \quad & f_1(x_1) + f_2(x_2) \\ \text{s.t.} \quad & A_1 x_1 = b_1, \\ & A_2 x_2 = b_2, \\ & x_1 - x_2 = 0, \\ & x_1 \in \mathcal{X}, \quad x_2 \in \mathcal{Y}, \end{aligned} \quad (4)$$

where $f_1(x_1) = f(x_1)$, $f_2(x_2) = f(x_2)$. An obvious advantage of such reformulation is that the given sets \mathcal{X} and \mathcal{Y} can be treated separately instead of regarding them as a whole. Given symmetric positive-definite matrices $P_1 \in \mathcal{R}^{m_1 \times m_1}$ and $P_2 \in \mathcal{R}^{m_2 \times m_2}$, the augmented Lagrangian function of (4) is given by

$$\begin{aligned} \mathcal{L}_\beta(x_1, x_2, \lambda) = & L(x_1, x_2, \lambda) \\ & + \frac{\beta}{2} \{ \|A_1 x_1 - b_1\|_{P_1}^2 + \|A_2 x_2 - b_2\|_{P_2}^2 + \|x_1 - x_2\|^2 \}, \end{aligned} \quad (5)$$

where $\beta > 0$ is a penalty parameter and

$$\begin{aligned} L(x_1, x_2, \lambda) = & f_1(x_1) + f_2(x_2) - \langle \lambda_1, A_1 x_1 - b_1 \rangle \\ & - \langle \lambda_2, A_2 x_2 - b_2 \rangle - \langle \lambda_3, x_1 - x_2 \rangle \end{aligned} \quad (6)$$

is the Lagrangian function of problem (4) with a multiplier $\lambda := (\lambda_1, \lambda_2, \lambda_3) \in \mathcal{R}^{m_1} \times \mathcal{R}^{m_2} \times \mathcal{R}^n$. Then, the introduced P-ADMM obeys the following iterative scheme:

$$\begin{aligned} x_1^{k+1} &= \arg \min_{x_1 \in \mathcal{X}} \mathcal{L}_\beta(x_1, x_2^k, \lambda^k) + \frac{\sigma_1 \beta}{2} \|x_1 - x_1^k\|^2, \\ x_2^{k+1} &= \arg \min_{x_2 \in \mathcal{Y}} \mathcal{L}_\beta(x_1^k, x_2, \lambda^k) + \frac{\sigma_2 \beta}{2} \|x_2 - x_2^k\|^2, \\ \lambda_1^{k+1} &= \lambda_1^k - \beta P_1 (A_1 x_1^{k+1} - b_1), \\ \lambda_2^{k+1} &= \lambda_2^k - \beta P_2 (A_2 x_2^{k+1} - b_2), \\ \lambda_3^{k+1} &= \lambda_3^k - \beta (x_1^{k+1} - x_2^{k+1}), \end{aligned} \quad (7)$$

where $\sigma_1 \in (1, +\infty)$, $\sigma_2 \in (1, +\infty)$ are two independent proximal parameters that control the proximity of the new iterative value to the last one; see, for example, [5] for more explanations.

Noting that scheme (7) can be regarded as an extended alternating direction method of multipliers (ADMM) of [6], because the two parameters σ_1 , σ_2 are independent instead of the same value and all the preconditioned matrices are not always identity matrices. For excellent reviews of the ADMM, we refer the reader to, for example, [6–14] and the

references therein, and also the recent published symmetric ADMM with larger step sizes in [15] which is an all-sided work for the two-block separable convex minimization problem. Besides, Goldstein et al. [16] developed a Nesterov's accelerated ADMM for problem (2) with partitions:

$$\begin{aligned} f(x) &= f_1(x_1) + f_2(x_2); \\ x &= (x_1, x_2)^T; \\ A &= [A_1, A_2], \end{aligned} \quad (8)$$

where the global convergence bounds were derived in terms of the dual objective to the original under the assumption that the two objectives were strongly convex. In 2016, He et al. [6] proved that both the Jacobian decomposition of the augmented Lagrangian method and the proximal point method were equivalent for solving the multiblock separable convex programming with one linear equality constraint, that is, problem (2) with partitions:

$$\begin{aligned} f(x) &= f_1(x_1) + \dots + f_m(x_m); \\ x &= (x_1, \dots, x_m)^T; \\ A &= [A_1, \dots, A_m]. \end{aligned} \quad (9)$$

Although there are lots of results about ADMM, most of concerned problems are still with one linear equality constraint, and research results of a preconditioned ADMM for (1) are few as far as we know.

Contributions of this paper are summarized as two aspects. One is that we introduce a novel preconditioned alternating direction method for solving the bilinear programming problem (1), and we prove that the constructed method is global convergent with a worst-case $\mathcal{O}(1/t)$ convergence rate in an ergodic sense. Another contribution is that several large-scale practical examples are tested to show the effectiveness of the proposed method. In Section 2, we analyze the convergence of the proposed method in detail. In Section 3, we first introduce a linearization technique for solving the involved subproblems of the P-ADMM (7) and then carry out some numerical experiments about the large-scale quadratic programming with two linear equality constraints. Finally, we conclude the paper in Section 4.

2. Convergence Analysis

By making use of the variational inequality, this section presents a unified framework to characterize the solution set of the reformulated problem (4) and the optimality conditions of the involved subproblems in (7). Moreover, the global convergence and the worst-case $\mathcal{O}(1/t)$ convergence rate of the proposed method are analyzed in detail. We begin with a basic lemma given in [15].

Lemma 1. *Let $f(x) : \mathcal{R}^m \rightarrow \mathcal{R}$ and $h(x) : \mathcal{R}^m \rightarrow \mathcal{R}$ be convex functions defined on a closed convex set $\Omega \subset \mathcal{R}^m$*

and $h(x)$ be differentiable. Assume that the solution set of the problem $\min_{x \in \Omega} \{f(x) + h(x)\}$ is nonempty. Then we have

$$\begin{aligned} x^* = \arg \min \{f(x) + h(x) \mid x \in \Omega\} &\iff x^* \in \Omega, \\ f(x) - f(x^*) + (x - x^*)^T \nabla h(x^*) &\geq 0, \quad \forall x \in \Omega. \end{aligned} \quad (10)$$

Any tuple $(x_1^*, x_2^*, \lambda^*)$ is called a saddle point of Lagrangian function (6) if it satisfies

$$L(x_1^*, x_2^*, \lambda) \leq L(x_1^*, x_2^*, \lambda^*) \leq L(x_1, x_2, \lambda^*), \quad (11)$$

which implies that finding a saddle point of $L(x_1, x_2, \lambda)$ is equivalent to finding a point

$$\begin{aligned} w^* &= (x_1^*, x_2^*, \lambda_1^*, \lambda_2^*, \lambda_3^*) \in \mathcal{M} \\ &= \mathcal{X} \times \mathcal{Y} \times \mathcal{R}^{m_1} \times \mathcal{R}^{m_2} \times \mathcal{R}^n \end{aligned} \quad (12)$$

such that

$$\begin{aligned} f_1(x_1) - f_1(x_1^*) + \langle x_1 - x_1^*, -A_1^T \lambda_1^* - \lambda_3^* \rangle &\geq 0, \\ \forall x_1 &\in \mathcal{X}, \\ f_2(x_2) - f_2(x_2^*) + \langle x_2 - x_2^*, -A_2^T \lambda_2^* + \lambda_3^* \rangle &\geq 0, \\ \forall x_2 &\in \mathcal{Y}, \\ \langle \lambda_1 - \lambda_1^*, A_1 x_1^* - b_1 \rangle &\geq 0, \\ \forall \lambda_1 &\in \mathcal{R}^{m_1}, \\ \langle \lambda_2 - \lambda_2^*, A_2 x_2^* - b_2 \rangle &\geq 0, \\ \forall \lambda_2 &\in \mathcal{R}^{m_2}, \\ \langle \lambda_3 - \lambda_3^*, x_1^* - x_2^* \rangle &\geq 0, \\ \forall \lambda_3 &\in \mathcal{R}^n. \end{aligned} \quad (13)$$

By rewriting the above inequalities as a compact variational inequality (VI), we have

$$\begin{aligned} \text{VI}(f, F, \mathcal{M}) : f(u) - f(u^*) + \langle w - w^*, F(w^*) \rangle \\ \geq 0, \quad \forall w \in \mathcal{M}, \end{aligned} \quad (14)$$

where

$$\begin{aligned} f(u) &= f_1(x_1) + f_2(x_2), \\ u &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \\ w &= \begin{pmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}, \\ F(w) &= \begin{pmatrix} -A_1^T \lambda_1 - \lambda_3 \\ -A_2^T \lambda_2 + \lambda_3 \\ A_1 x_1 - b_1 \\ A_2 x_2 - b_2 \\ x_1 - x_2 \end{pmatrix}. \end{aligned} \quad (15)$$

Note that the mapping $F(w)$ is skew-symmetric, so the following fundamental property holds:

$$\langle w - \widehat{w}, F(w) - F(\widehat{w}) \rangle = 0, \quad \forall w, \widehat{w} \in \mathcal{M}. \quad (16)$$

By the assumption that the solution set of (1) is nonempty, the solution set \mathcal{M}^* of VI(f, F, \mathcal{M}) is also nonempty. The next theorem describes a concrete way to characterizing the set \mathcal{M}^* , whose proof is the same as that of Theorem 2 [10] and is omitted here.

Theorem 2. *The solution set of VI(f, F, \mathcal{M}) in (14) is convex and can be expressed as*

$$\begin{aligned} \mathcal{M}^* &= \bigcap_{w \in \mathcal{M}} \{ \widehat{w} \in \mathcal{M} \mid f(u) - f(\widehat{u}) + \langle w - \widehat{w}, F(w) \rangle \\ &\geq 0 \}. \end{aligned} \quad (17)$$

For any $w \in \mathcal{H}(\widehat{w}) = \{w \in \mathcal{M} \mid \|w - \widehat{w}\| \leq 1\}$, Theorem 2 shows that if

$$\sup \{ f(\widehat{u}) - f(u) + \langle \widehat{w} - w, F(w) \rangle \} \leq \epsilon, \quad (18)$$

then the vector $\widehat{w} \in \mathcal{M}$ is called an ϵ -approximate solution of VI(f, F, \mathcal{M}), where $\epsilon > 0$ is an accuracy, especially $\epsilon = \mathcal{O}(1/t)$.

Lemma 3. *Let the sequence $\{w^{k+1}\}$ be generated by the algorithm P-ADMM. Then we have*

$$\begin{aligned} f(u) - f(u^{k+1}) \\ + \langle w - w^{k+1}, F(w^{k+1}) \rangle + G(w^{k+1} - w^k) &\geq 0, \\ \forall w \in \mathcal{M}, \end{aligned} \quad (19)$$

where

$$G = \begin{bmatrix} \sigma_1 \beta I & \beta I & 0 & 0 & 0 \\ \beta I & \sigma_2 \beta I & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\beta} P_1^{-1} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} P_2^{-1} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\beta} I \end{bmatrix}, \quad (20)$$

$$w^k = \begin{pmatrix} x_1^k \\ x_2^k \\ \lambda_1^k \\ \lambda_2^k \\ \lambda_3^k \end{pmatrix}.$$

Proof. Applying Lemma 1, the optimality conditions of the two subproblems in (7) are

$$\begin{aligned} & f_1(x_1) - f_1(x_1^{k+1}) + \langle x_1 - x_1^{k+1}, -A_1^T \lambda_1^k - \lambda_3^k \\ & + \beta A_1^T P_1 (A_1 x_1^{k+1} - b_1) + \beta (x_1^{k+1} - x_2^k) \\ & + \sigma_1 \beta (x_1^{k+1} - x_1^k) \rangle \geq 0, \quad \forall x_1 \in \mathcal{X}, x_1^{k+1} \in \mathcal{X}, \\ & f_2(x_2) - f_2(x_2^{k+1}) + \langle x_2 - x_2^{k+1}, -A_2^T \lambda_2^k + \lambda_3^k \\ & + \beta A_2^T P_2 (A_2 x_2^{k+1} - b_2) - \beta (x_1^k - x_2^{k+1}) \\ & + \sigma_2 \beta (x_2^{k+1} - x_2^k) \rangle \geq 0, \quad \forall x_2 \in \mathcal{Y}, x_2^{k+1} \in \mathcal{Y}. \end{aligned} \quad (21)$$

Since the update of the Lagrange multipliers in (7) satisfies

$$\begin{aligned} \lambda_1^k &= \lambda_1^{k+1} - \beta P_1 (A_1 x_1^{k+1} - b_1), \\ \lambda_2^k &= \lambda_2^{k+1} - \beta P_2 (A_2 x_2^{k+1} - b_2), \\ \lambda_3^k &= \lambda_3^{k+1} - \beta (x_1^{k+1} - x_2^{k+1}), \end{aligned} \quad (22)$$

substituting (22) into (21) we obtain

$$\begin{aligned} & f_1(x_1) - f_1(x_1^{k+1}) + \langle x_1 - x_1^{k+1}, -A_1^T \lambda_1^{k+1} - \lambda_3^{k+1} \\ & + \sigma_1 \beta (x_1^{k+1} - x_1^k) + \beta (x_2^{k+1} - x_2^k) \rangle \geq 0, \\ & f_2(x_2) - f_2(x_2^{k+1}) + \langle x_2 - x_2^{k+1}, -A_2^T \lambda_2^{k+1} + \lambda_3^{k+1} \\ & + \beta (x_1^{k+1} - x_1^k) + \sigma_2 \beta (x_2^{k+1} - x_2^k) \rangle \geq 0. \end{aligned} \quad (23)$$

Notice that (22) can be rewritten as

$$\begin{aligned} & \left\langle \lambda_1 - \lambda_1^{k+1}, A_1 x_1^{k+1} - b_1 + \frac{1}{\beta} P_1^{-1} (\lambda_1^{k+1} - \lambda_1^k) \right\rangle, \\ & \quad \forall \lambda_1 \in \mathcal{R}^{m_1}, \lambda_1^{k+1} \in \mathcal{R}^{m_1}, \\ & \left\langle \lambda_2 - \lambda_2^{k+1}, A_2 x_2^{k+1} - b_2 + \frac{1}{\beta} P_2^{-1} (\lambda_2^{k+1} - \lambda_2^k) \right\rangle, \\ & \quad \forall \lambda_2 \in \mathcal{R}^{m_2}, \lambda_2^{k+1} \in \mathcal{R}^{m_2}, \\ & \left\langle \lambda_3 - \lambda_3^{k+1}, x_1^{k+1} - x_2^{k+1} + \frac{1}{\beta} (\lambda_3^{k+1} - \lambda_3^k) \right\rangle, \\ & \quad \forall \lambda_3 \in \mathcal{R}^n, \lambda_3^{k+1} \in \mathcal{R}^n. \end{aligned} \quad (24)$$

Combining (23) and (24), we immediately complete the proof. \square

Note that matrix G in Lemma 3 is strictly SPD, because the upper-left 2×2 block matrix is SPD for any $\sigma_1 > 1$, $\sigma_2 > 1$ and the lower-right 3×3 diagonal matrix is SPD from the symmetric positivity of the matrices P_1, P_2 and $\beta > 0$. Compared with the inequalities (14) and (19), the key to prove the convergence of the algorithm P-ADMM is to verify that the cross term of (19) converges to zero, that is,

$$\lim_{k \rightarrow \infty} \langle w - w^{k+1}, G(w^{k+1} - w^k) \rangle = 0, \quad \forall w \in \mathcal{M}. \quad (25)$$

In other words, the sequence $\{w^k - w^*\}$ would be contractive under the weighted matrix G . In what follows, we will show such assertion by using the definition $\|w\|_G = \sqrt{w^T G w}$ for any $w \in \mathcal{R}^{m_1+m_2+3n}$.

Lemma 4. *The sequence $\{w^{k+1}\}$ generated by the algorithm P-ADMM satisfies*

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - \|w^k - w^{k+1}\|_G^2. \quad (26)$$

Proof. Setting $w = w^*$ in (19), it follows that

$$\begin{aligned} & \langle w^* - w^{k+1}, G(w^{k+1} - w^k) \rangle \\ & \geq f(u^{k+1}) - f(u^*) + \langle w^{k+1} - w^*, F(w^{k+1}) \rangle. \end{aligned} \quad (27)$$

By making use of (16) and (14), we get

$$\begin{aligned} & f(u^{k+1}) - f(u^*) + \langle w^{k+1} - w^*, F(w^{k+1}) \rangle \\ & \geq f(u^{k+1}) - f(u^*) + \langle w^{k+1} - w^*, F(w^*) \rangle \geq 0, \end{aligned} \quad (28)$$

which leads to

$$\langle w^{k+1} - w^*, G(w^k - w^{k+1}) \rangle \geq 0. \quad (29)$$

Based on (29) and the symmetric positivity of G , we can obtain

$$\begin{aligned} \|w^k - w^*\|_G^2 &= \|w^k - w^{k+1} + w^{k+1} - w^*\|_G^2 \\ &= \|w^k - w^{k+1}\|_G^2 + \|w^{k+1} - w^*\|_G^2 \\ &\quad + 2 \langle w^{k+1} - w^*, G(w^k - w^{k+1}) \rangle \\ &\geq \|w^k - w^{k+1}\|_G^2 + \|w^{k+1} - w^*\|_G^2. \end{aligned} \quad (30)$$

□

Theorem 5. Let the sequence $\{w^{k+1}\}$ be generated by the algorithm P-ADMM, then the following assertions hold:

- (a) $\lim_{k \rightarrow \infty} (w^k - w^{k+1}) = 0$.
- (b) The sequence $\{w^{k+1}\}$ is bounded.
- (c) Any accumulation point of $\{w^{k+1}\}$ is a solution point of $VI(f, F, \mathcal{M})$.
- (d) There exists $w^\infty \in \mathcal{M}^*$ such that $\lim_{k \rightarrow \infty} w^{k+1} = w^\infty$.

Proof. Summing the inequality (26) over $k = 0, 1, 2, \dots, \infty$, we have

$$\sum_{k=0}^{\infty} \|w^k - w^{k+1}\|_G^2 \leq \|w^0 - w^*\|_G^2, \quad (31)$$

which implies $\lim_{k \rightarrow \infty} (w^k - w^{k+1}) = 0$ because of the symmetric positivity of the matrix G . The assertion (b) is evident followed by (a). By taking the limit of (19) and using the assertion (a), we get

$$\lim_{k \rightarrow \infty} \{f(u) - f(u^{k+1}) + \langle w - w^{k+1}, F(w^{k+1}) \rangle\} \geq 0, \quad (32)$$

$\forall w \in \mathcal{M}$,

which shows that $\lim_{k \rightarrow \infty} w^{k+1}$ is a solution point of $VI(f, F, \mathcal{M})$, that is, the assertion (c) holds.

Let w^∞ be an accumulation point of $\{w^{k+1}\}$. Then the third assertion implies that $w^\infty \in \mathcal{M}^*$ and

$$\|w^{k+1} - w^\infty\|_G^2 \leq \|w^k - w^\infty\|_G^2 - \|w^k - w^{k+1}\|_G^2. \quad (33)$$

Using the above inequality together with the assertion (a), the proof of (d) is completed. □

Theorem 6. For any integer $t > 0$ and the sequence $\{w^{k+1}\}$ generated by the P-ADMM (7), let

$$\widehat{w}_t = \frac{1}{t+1} \sum_{k=0}^t w^{k+1}. \quad (34)$$

Then it holds that

$$\begin{aligned} &f(\widehat{w}_t) - f(u) + \langle \widehat{w}_t - w, F(w) \rangle \\ &\leq \frac{1}{2(t+1)} \|w^0 - w\|_G^2, \quad \forall w \in \mathcal{M}. \end{aligned} \quad (35)$$

Proof. Clearly, $\widehat{w}_t \in \mathcal{M}$ since it can be treated as a convex combination of w^{k+1} ($k = 0, 1, \dots, t$). Substituting (16) into (19), we deduce that

$$\begin{aligned} &f(u) - f(u^{k+1}) + \langle w - w^{k+1}, F(w) \rangle \\ &\geq \langle w^{k+1} - w, G(w^{k+1} - w^k) \rangle, \quad \forall w \in \mathcal{M}. \end{aligned} \quad (36)$$

By utilizing an identity

$$\begin{aligned} 2 \langle a - b, G(c - d) \rangle &= \|a - d\|_G^2 - \|a - c\|_G^2 + \|c - b\|_G^2 \\ &\quad - \|d - b\|_G^2, \end{aligned} \quad (37)$$

we obtain

$$\begin{aligned} &\langle w^{k+1} - w, G(w^{k+1} - w^k) \rangle \\ &= \frac{1}{2} (\|w^{k+1} - w\|_G^2 + \|w^{k+1} - w^k\|_G^2 - \|w^k - w\|_G^2) \\ &\geq \frac{1}{2} (\|w^{k+1} - w\|_G^2 - \|w^k - w\|_G^2), \end{aligned} \quad (38)$$

which makes (36) become

$$\begin{aligned} &f(u) - f(u^{k+1}) + \langle w - w^{k+1}, F(w) \rangle + \frac{1}{2} \|w^k - w\|_G^2 \\ &\geq \frac{1}{2} \|w^{k+1} - w\|_G^2. \end{aligned} \quad (39)$$

Summing the above inequality over $k = 0, 1, \dots, t$, we have

$$\begin{aligned} &(t+1)f(u) - \sum_{k=0}^t f(u^{k+1}) \\ &+ \left\langle (t+1)w - \sum_{k=0}^t w^{k+1}, F(w) \right\rangle + \frac{1}{2} \|w^0 - w\|_G^2 \\ &\geq 0 \iff \end{aligned} \quad (40)$$

$$\begin{aligned} &\frac{1}{t+1} \sum_{k=0}^t f(u^{k+1}) - f(u) + \langle \widehat{w}_t - w, F(w) \rangle \\ &\leq \frac{1}{2(t+1)} \|w^0 - w\|_G^2. \end{aligned}$$

Since $f(u)$ is convex and $\widehat{u}_t = (1/(t+1)) \sum_{k=0}^t u^{k+1}$, so it holds that

$$f(\widehat{u}_t) \leq \frac{1}{t+1} \sum_{k=0}^t f(u^{k+1}). \quad (41)$$

Substituting it into (40), the proof is completed. □

Remark 7. Theorem 5 shows that the P-ADMM (7) is globally convergent. And Theorem 6 tells us that for any given compact set $\mathcal{K} \subset \mathcal{M}$ and $\eta := \sup_{w \in \mathcal{K}} \|w^0 - w\|_G^2$, the vector \widehat{w}_t must satisfy

$$\sup_{w \in \mathcal{K}} \{f(\widehat{w}_t) - f(u) + \langle \widehat{w}_t - w, F(w) \rangle\} \leq \frac{\eta}{2(t+1)}, \quad (42)$$

which shows that the proposed method converges in a worst-case $\mathcal{O}(1/t)$ rate in an ergodic sense.

Remark 8. The penalty parameter β in (7) can be updated by the formula $\beta^{k+1} = \tau\beta^k$ with $\tau > 0$, and it is a constant when taking $\tau = 1$. The preconditioned matrices P_1 and P_2 are usually chosen as the identity matrix, the diagonal matrix with positive diagonal entries, or the tridiagonal matrix.

3. Numerical Experiments

In this section, we investigate the feasibility and efficiency of the proposed method by some numerical experiments about the quadratic programming model with two linear equality constraints. The codes of the algorithm P-ADMM are written in MATLAB 7.10 (R2010a) and the experiments are carried out on a PC with Intel Core i5 processor (3.3 GHz) with 4 GB memory. Inspired by Theorem 5, we take an easily

implementable stopping criterion for the proposed method, that is,

$$\text{ERR}(k) = \max \left\{ \|x_1^k - x_1^{k+1}\|_{\infty}, \|x_2^k - x_2^{k+1}\|_{\infty} \right\} \leq \text{tol}, \quad (43)$$

where tol is the given tolerance and x_i^k is the k th iteration generated by scheme (7).

To avoid the case that the subproblems of (7) have no explicit solution, the two preconditioned matrices are simply chosen as the identity matrix, and we then can use a linearized strategy to accelerate the convergence of solving the subproblems. Without loss of generality, we take the x_1 -subproblem, for example. In such case, the x_1 -subproblem is equivalent to

$$\begin{aligned} x_1^{k+1} &= \arg \min_{x_1 \in \mathcal{X}} \left\{ f_1(x_1) + \frac{\beta}{2} \left(\left\| A_1 x_1 - b_1 - \frac{\lambda_1^k}{\beta} \right\|^2 + \left\| x_1 - x_2^k - \frac{\lambda_3^k}{\beta} \right\|^2 + \sigma_1 \|x_1 - x_1^k\|^2 \right) \right\} \\ &= \arg \min_{x_1 \in \mathcal{X}} \left\{ f_1(x_1) + \frac{\beta}{2} \|Ax_1 - \mathbf{a}\|^2 \right\}, \end{aligned} \quad (44)$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} A_1 \\ I \\ \sqrt{\sigma_1} I \end{bmatrix} \in \mathcal{R}^{(m_1+2n) \times n}, \\ \mathbf{a} &= \begin{pmatrix} b_1 + \frac{\lambda_1^k}{\beta} \\ x_2^k + \frac{\lambda_3^k}{\beta} \\ \sqrt{\sigma_1} x_1^k \end{pmatrix} \in \mathcal{R}^{m_1+2n}. \end{aligned} \quad (45)$$

By the well-known Taylor formula in mathematical analysis, the quadratic term $\|Ax_1 - \mathbf{a}\|^2/2$ can be approximated by

$$\begin{aligned} \frac{1}{2} \|Ax_1 - \mathbf{a}\|^2 &\approx \frac{1}{2} \|Ax_1^k - \mathbf{a}\|^2 + \langle g^k, x_1 - x_1^k \rangle \\ &\quad + \frac{1}{2\eta} \|x_1 - x_1^k\|^2, \end{aligned} \quad (46)$$

where $\eta > 0$ is a proximal factor and $g^k = \mathbf{A}^T(\mathbf{A}x_1^k - \mathbf{a})$ is the gradient of the quadratic term at x_1^k . Hence, the objective function in (44) is of the equivalent form:

$$f_1(x_1) + \frac{\beta}{2\eta} \|x_1 - x_1^k - \eta g_1^k\|^2, \quad (47)$$

which makes the x_1 -subproblem have closed solution form. The x_2 -subproblem can be also tackled in a similar way as the x_1 -subproblem. For more cases that are analogous to (47), the explicit solution form can be dated back to Lemmas 1 and 3 given in [17].

In what follows, the penalty parameter β is updated by the formula $\beta^{k+1} = 5\beta^k$ with $\beta^0 = 0.5e - 3$, the proximal parameters are chosen as $(\sigma_1, \sigma_2) = (2, 2)$, $\eta = 0.25e - 2$, and the iterative variables are initialized as fixed values $(x_1^0, x_2^0) = (\text{ones}(n, 1), \text{ones}(n, 1))$. Generally speaking, the quadratic programming with two linear equality constraints is of the following form:

$$\begin{aligned} \min \quad & \frac{1}{2} x^T H x + c^T x \\ \text{s.t.} \quad & A_1 x = b_1, \\ & A_2 x = b_2, \\ & x \in \Omega, \end{aligned} \quad (48)$$

where $H \in \mathcal{R}^{n \times n}$ is a positive semidefinite matrix. Model (48) includes the classic Markowitz portfolio optimization problem as a special case; see, for example, [2] and Example 10.

Example 9. Consider model (48) with a simple case $\Omega = \mathcal{R}^n$, where the given data are randomly generated by the following MATLAB codes:

$$\begin{aligned} H_1 &= \text{randn}(n); \\ H &= H_1^T * H_1; \\ c &= \text{rand}(n, 1); \\ A_1 &= \text{randn}(n); \\ A_2 &= \text{randn}(n); \\ b_1 &= \text{rand}(n, 1); \\ b_2 &= \text{rand}(n, 1). \end{aligned} \quad (49)$$

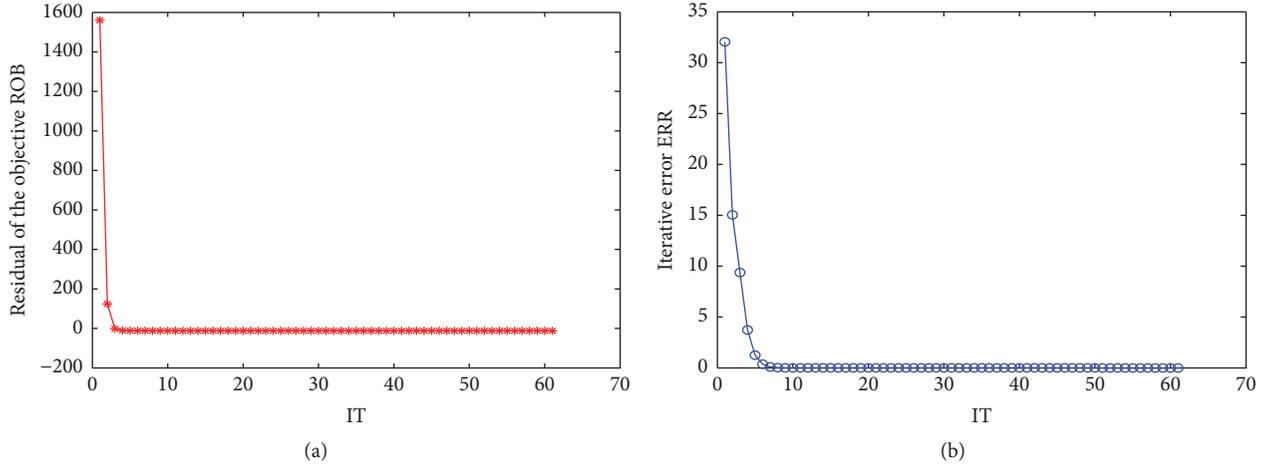


FIGURE 1: Convergence curves of the residual ROB (a) and the iterative error ERR (b).

TABLE 1: Experimental results of Example 9 by the P-ADMM.

tol	IT	CPU	ERR	ROB
$1e-1$	8	0.0265	0.0273	-11.9488
$1e-3$	12	0.0593	$9.7241e-4$	-12.0118
$1e-5$	19	0.0518	$7.7188e-6$	-12.0160
$1e-7$	26	0.0646	$6.0438e-8$	-12.0161
$1e-9$	32	0.0846	$9.4608e-10$	-12.0161
$1e-11$	39	0.0790	$7.4072e-12$	-12.0161
$1e-13$	46	0.1039	$5.7954e-14$	-12.0161
$1e-15$	61	0.1232	$8.8818e-16$	-12.0161

For this example, Table 1 reports several experimental results of Example 9 with $n = 200$ by the algorithm P-ADMM with different tolerance error, including the number of iterations (denoted by “IT”), the CPU time (denoted by “CPU”), the iterative error of the solution (denoted by “ERR”), and the residual of the objective (denoted by “ROB”). Figure 1 still draws the convergence curves of the residual of the objective and the iterative error of the solution under the tolerance $\text{tol} = 1.0 \times 10^{-15}$, respectively.

From Table 1, we can see that when using the P-ADMM to solve Example 9, the CPU time cost is less than 0.2 seconds and the number of the iterations is not bigger than 65 steps. The obtained results, including the residual error ERR listed in Table 1 and the convergence curves depicted in Figure 1, verify the feasibility and efficiency of the P-ADMM scheme for solving the small-scale problem. Besides, the last two columns of Table 1 imply that we can choose a relatively tiny stopping criterion (e.g., $\text{tol} = 1.0 \times 10^{-5}$) to obtain nearly the same value of the objective and to save the CPU time.

Example 10. Consider model (48) with $\Omega = \{x \in \mathcal{R}^n \mid x \geq 0\}$ and

$$\begin{aligned}
 c &= \text{zeros}(n, 1); \\
 A_1 &= \text{ones}(1, n); \\
 b_1 &= 1; \\
 A_2 &= r \in \mathcal{R}^{1 \times n}; \\
 b_2 &= p \in \mathcal{R}^+.
 \end{aligned} \tag{50}$$

Then model (48) immediately becomes the Markowitz portfolio optimization problem:

$$\begin{aligned}
 \min \quad & \frac{1}{2} x^T H x \\
 \text{s.t.} \quad & e^T x = 1, \\
 & r^T x = p, \\
 & x \geq 0,
 \end{aligned} \tag{51}$$

where the matrix $H \in \mathcal{R}^{n \times n}$ stands for the covariance matrix of the return on the n assets in the portfolio, the variable x denotes the vector of portfolio weights that represent the amount of capital to be invested in each asset, r is the vector of expected returns of the different assets, and p is a given total return. In such case, the solution of (51) is sparse, which also verifies why some researchers try to find the sparse solution of the problem (51), see, for example, [18].

For Example 10, we test eleven large-scale experiments, in which matrix H is generated in the same way as that of Example 9 and r, p are, respectively, generated by the MATLAB inner functions $\text{rand}(1, n)$ and $\text{rand}(1, 1)$. Table 2 reports some experimental results of this example with different dimension $n \in [800, 2800]$, where the tolerance error of the algorithm is set as $\text{tol} = 10^{-5}$. The notations IT, CPU, ERR, and ROB are the same meanings as mentioned in Example 9, the number of the nonzero entries of the solution x^* is denoted by $\|x^*\|_0$, and the sparsity ratio is defined as $\|x^*\|_0/n \times 100\%$. The convergence curves of the residual of the objective and the iterative error of the solution for Example 10 with $n \in [800, 2800]$ are depicted in Figure 2.

An outstanding observation from Table 2 is that both the number of the iteration (< 25) and the CPU time (< 16 s) are small, and the CPU time increases along with the increase of the dimension n of x . Another observation is that the sparsity ratio of the solution is over 50%, which implies that more than half of the assets are not necessary to be invested and also provides some useful suggestions for an investor in finance. Both Table 2 and Figure 2 show that the P-ADMM

TABLE 2: Experimental results of Example 10 by P-ADMM.

n	IT	CPU	ERR	ROB	$\ x^*\ _0$	$\ x^*\ _0/n$
800	19	0.7516	$9.5680e-6$	$2.5181e-4$	444	55.50%
1000	19	1.2269	$8.4815e-6$	$1.7696e-4$	548	54.80%
1200	20	2.0222	$7.4162e-6$	$3.1982e-4$	651	54.25%
1400	22	3.4151	$5.8193e-6$	$4.3066e-4$	792	56.57%
1600	20	4.3842	$6.1633e-6$	0.0089	877	54.81%
1800	19	5.4484	$7.5718e-6$	0.0015	940	52.22%
2000	19	6.8253	$8.5260e-6$	0.0207	1070	53.50%
2200	19	8.8434	$9.5873e-6$	0.0039	1180	53.64%
2400	20	11.1692	$9.6233e-6$	0.0482	1302	54.25%
2600	20	14.0278	$5.9587e-6$	0.0289	1387	53.35%
2800	18	15.5509	$6.4949e-6$	0.0067	1507	53.82%

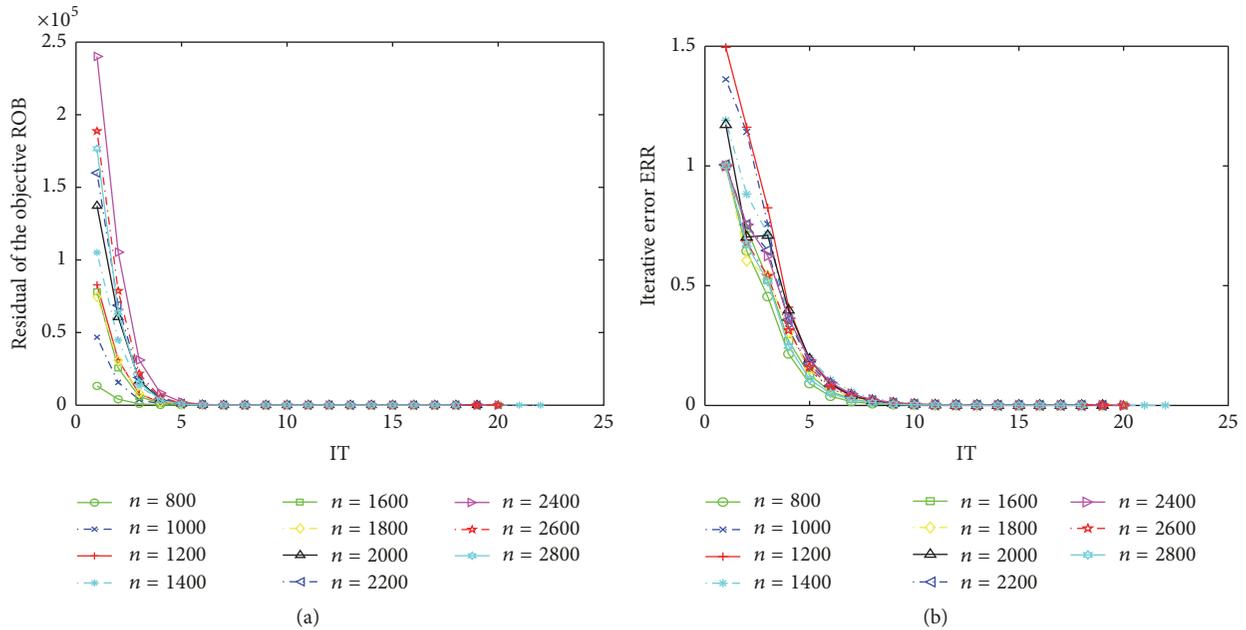


FIGURE 2: Convergence curves of the residual ROB (a) and the iterative error ERR (b).

(7) is robust for solving the large-scale Markowitz portfolio optimization problem.

4. Conclusion

Instead of studying the optimization problem with one linear constraint, in this paper, we concentrate our attentions on the generalized bilinear programming problem and develop a preconditioned alternating direction method. Based on the traditional proof of the ADMM, the global convergence of the proposed method is proved and the worst-case $\mathcal{O}(1/t)$ convergence rate in an ergodic sense is established. In order to avoid the case that the subproblem has no explicit solution, we still use a linearized strategy to approximately tackle the involved subproblems in the proposed method. Numerical results show that the proposed method is feasible and efficient.

Nowadays, many researchers are interested in the ADMM which can be regarded as an alternating update method

for the variables and Lagrange multipliers for the separable convex programming. For the nonseparable convex optimization problem, the famous Taylor formula motivates us to use the first-order approximation to linearly deal with the objective function of the problem, and then one can design the corresponding ADMM to solve it. Noticing that the proposed method in current paper can be applied to the above scenarios and can be also used to solve the matrix minimization problem with two linear constraints.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] H. Konno, P. T. Thach, and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, Springer, Dordrecht, Netherland, 1997.

- [2] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investment*, John Wiley and Sons, New York, NY, USA, 1959.
- [3] J.-J. Moreau, "Proximite et dualite dans un espace hilbertien," *Bulletin of the French Mathematical Society*, vol. 93, pp. 273–299, 1965.
- [4] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969.
- [5] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976.
- [6] B. He, H.-K. Xu, and X. Yuan, "On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 1204–1217, 2016.
- [7] J. C. Bai, H. C. Zhang, J. C. Li, and F. M. Xu, "Generalized symmetric alternating direction method for separable convex programming," <http://www.optimization-online.org/DB-FILE/2016/10/5699.pdf>.
- [8] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.
- [9] E. X. Fang, B. He, H. Liu, and X. Yuan, "Generalized alternating direction method of multipliers: new theoretical insights and applications," *Mathematical Programming Computation*, vol. 7, no. 2, pp. 149–187, 2015.
- [10] B. He and X. Yuan, "On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.
- [11] B. He and X. Yuan, "Block-wise alternating direction method of multipliers for multiple-block convex programming and beyond," *SMAI Journal of Computational Mathematics*, vol. 1, pp. 145–174, 2015.
- [12] D. Sun, K.-C. Toh, and L. Yang, "A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 882–915, 2015.
- [13] J. J. Wang and W. Song, "An algorithm twisted from generalized ADMM for multi-block separable convex minimization models," *Journal of Computational and Applied Mathematics*, vol. 309, pp. 342–358, 2017.
- [14] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009.
- [15] B. He, F. Ma, and X. Yuan, "Convergence study on the symmetric version of ADMM with larger step sizes," *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 1467–1501, 2016.
- [16] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.
- [17] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.
- [18] X. Chen and S. Xiang, "Sparse solutions of linear complementarity problems," *Mathematical Programming*, vol. 159, no. 1-2, Ser. A, pp. 539–556, 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

