J. Nonlinear Var. Anal. 6 (2022), No. 6, pp. 707-723
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.6.2022.6.08

A NEW ACCELERATED POSITIVE-INDEFINITE PROXIMAL ADMM FOR CONSTRAINED SEPARABLE CONVEX OPTIMIZATION PROBLEMS

JIE LIU[1], JIAWEI CHEN[1,*], JINLAN ZHENG[1], XUERUI ZHANG[2], ZHONGPING WAN[3]

[1]*School of Mathematics and statistics, Southwest University, Chongqing 400715, China*
[2]*School of Management, Chongqing College of Humanities, Science & Technology, Chongqing 401524, China*
[3]*School of Mathematics and statistics, Wuhan University, Wuhan 430072, China*

Abstract. The alternating direction method of multipliers (ADMM) is a powerful method to solve constrained convex optimization problems with the separable structure. The ADMM with the positive-indefinite proximal terms, which has ergodic convergent rate $O(\frac{1}{K})$ with the number of iterations K, is more general than the ADMM with positive-definite proximal terms. In this paper, we propose a new accelerated positive-indefinite proximal linearized ADMM algorithm with positive-indefinite proximal matrix by the techniques of extrapolation. We obtain the nonergodic convergence rate $O(\frac{1}{K})$ in the sense of objective values and the nonergodic convergence rate $O(\frac{1}{\sqrt{K}})$ in the sense of iterative sequence of the proposed method as well as the upper bound of the violation of constraints. Numerical results are reported to show the efficiency of the proposed method.

Keywords. ADMM; Nonergodic convergence rate; Positive-indefinite proximal linearized ADMM; Separable convex optimization; Violation of constraints.

1. INTRODUCTION

In this paper, we consider the following separable convex minimization problem:

$$\min\{f(x)+g(y)\,|\,Ax+By=b,x\in\mathscr{X},y\in\mathscr{Y}\},\tag{1.1}$$

where $A\in\mathbb{R}^{m\times n_1}$, $B\in\mathbb{R}^{m\times n_2}$, $b\in\mathbb{R}^m$, $\mathscr{X}\subseteq\mathbb{R}^{n_1}$, and $\mathscr{Y}\subseteq\mathbb{R}^{n_2}$ are nonempty closed convex sets, $f:\mathbb{R}^{n_1}\to\mathbb{R}\cup\{+\infty\}$ and $g:\mathbb{R}^{n_2}\to\mathbb{R}\cup\{+\infty\}$ are two proper convex and lower semicontinuous (not necessarily smooth) functions. Problem (1.1) is assumed to have a solution in this paper.

The augmented Lagrangian function of problem (1.1) is defined as:

$$\mathscr{L}_\beta(x,y,\lambda):=f(x)+g(y)-\lambda^\top(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|^2,\tag{1.2}$$

where $\lambda\in\mathbb{R}^m$ is the Lagrangian multiplier, and $\beta>0$ is a penalty parameter.

The classic alternating direction method of multipliers (shortly, ADMM), which is closely related to the Douglas-Rachford split algorithm, was first proposed by Gabay, Mercier and

*Corresponding author.
E-mail addresses: liujie66j@163.com (J. Liu), j.w.chen713@163.com (J. Chen), zjlswu1234@163.com (J. Zheng), leegohi1993@gmail.com (X. Zhang), zpwan-whu@126.com (Z. Wan).

Received July 11, 2021; Accepted June 10, 2022.

Glovinski in the mid-1970s [1, 2]. For a given iterate (y_k, λ_k), the iterative scheme of the ADMM for solving problem (1.1) is as follows:

$$\begin{cases} x_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x, y_k, \lambda_k) \,|\, x \in \mathscr{X}\right\}, \\ y_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x_{k+1}, y, \lambda_k) \,|\, y \in \mathscr{Y}\right\}, \\ \lambda_{k+1} = \lambda_k - \beta(Ax_{k+1} + By_{k+1} - b). \end{cases} \tag{1.3}$$

The ergodic convergence rate of the ADMM algorithm was proved to be $o(\frac{1}{K})$ in [3], and the non-ergodic convergence rate was also proved to be $o(\frac{1}{\sqrt{K}})$ in [4]. The ADMM was successfully applied in various fields, such as statistical learning, computer vision, image processing, wireless network, and distributed network; see, e.g., [5, 6, 7, 8, 9, 10, 11, 12] and the references therein.

The key point of dealing with ADMM (1.3) is to solve the subproblem to obtain the iteration point (x_{k+1}, y_{k+1}). An effective tactics is to regularize the subproblem by a quadratic proximal term. Without loss of generality, we just discuss the case where only the y-subproblem of the ADMM (1.3) is regularized as follows (Generally, it is also probable to consider the situation when the x-subproblem and y-subproblem are both proximally regularized; see [13].):

$$(\mathrm{PD-ADMM}) \begin{cases} x_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x, y_k, \lambda_k) \,|\, x \in \mathscr{X}\right\}, \\ y_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x_{k+1}, y, \lambda_k) + \frac{1}{2}\|y - y_k\|_D^2 \,|\, y \in \mathscr{Y}\right\}, \\ \lambda_{k+1} = \lambda_k - \beta\left(Ax_{k+1} + By_{k+1} - b\right). \end{cases} \tag{1.4}$$

In (1.4), $\frac{1}{2}\|y - y_k\|_D^2$ is the quadratic proximal regularization term and $D \in \mathbb{R}^{n_2 \times n_2}$ is the proximal matrix that is always required to be positive definite in the literature. PD-ADMM (1.4) is widely used in various fields, such as image processing, compressive sensing, statistical learning and so on; see, e.g., [14, 15] and the references therein.

Recently, a possibility of relaxing the positive definiteness requirement of the PD-ADMM was presented, which implies that it was not necessary to require the positive definiteness of the proximal matrix in [16]. Motivated by [16], a new positive-indefinite proximal linearized ADMM (PIPL-ADMM) was proposed to solve problem(1.1) in [17] as follows:

$$\begin{cases} x_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x, y_k, \lambda_k) \,|\, x \in \mathscr{X}\right\}, \\ y_{k+1} = \arg\min\left\{\mathscr{L}_\beta(x_{k+1}, y, \lambda_k) + \frac{1}{2}\|y - y_k\|_{D_0}^2 \,|\, y \in \mathscr{Y}\right\}, \\ \lambda_{k+1} = \lambda_k - \gamma\beta(Ax_{k+1} + By_{k+1} - b), , \end{cases} \tag{1.5}$$

where $D_0 = D - (1-\tau)\beta B^\top B$ with an arbitrarily given positive-definite matrix $D \in \mathbb{R}^{n_2 \times n_2}$ and $\tau \in (0, 1]$, $\gamma \in (0, \frac{\sqrt{5}+1}{2})$. The PIPL-ADMM algorithm relaxed the restriction on the positive definiteness of the neighbor matrix, and it was proved that the ergodic convergence rate is $o(1/k)$, where k is the number of iterations. It is precisely because the PIPL-ADMM algorithm has more relaxed conditions and is more in line with the reality of life, so it has a wider application in real life.

To speed up computations, the basic algorithm is generally accelerated, usually starting from two directions: reducing the number of iterations or shortening the running time. In [18], Li et al. simplified and improved the acceleration algorithm proposed by Ouyang et al. [19] to accelerate the ADMM algorithm through Nesterov's second acceleration scheme, and the new acceleration algorithm is as follows:

$$\begin{cases} \frac{1-\theta_k}{\theta_k} = \frac{1}{\theta_{k-1}} - \tau, \\ v_k = y_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(y_k - y_{k-1}), \\ x_{k+1} = \arg\min_x \left\{ f(x) + \langle \lambda_k, Ax \rangle + \frac{\beta}{2\theta_k}\|Ax + Bv_k - b\|^2 \right\}, \\ y_{k+1} = \arg\min_y \left\{ g(y) + \langle \lambda_k, By \rangle + \frac{\beta}{2\theta_k}\|Ax_{k+1} + By - b\|^2 \right\}, \\ \lambda_{k+1} = \lambda_k + \beta\gamma(Ax_{k+1} + By_{k+1} - b). \end{cases} \tag{1.6}$$

Inspired by the works [17, 18, 19], this paper proposes an accelerated PIPL-ADMM (APIPL-ADMM) algorithm by using the accelerated technique [18, 19] as follows:

$$\begin{cases} \frac{1-\theta_k}{\theta_k} = \frac{1}{\theta_{k-1}} - \gamma, \\ v_k = y_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(y_k - y_{k-1}), \\ x_{k+1} = \arg\min_x \left\{ f(x) + \langle \lambda_k, Ax \rangle + \frac{\beta}{2\theta_k}\|Ax + Bv_k - b\|^2 \right\}, \\ y_{k+1} = \arg\min_y \left\{ g(y) + \langle \lambda_k, By \rangle + \frac{\beta}{2\theta_k}\|Ax_{k+1} + By - b\|^2 + \frac{1}{2}\|y - y_k\|^2_{D_0} \right\}, \\ \lambda_{k+1} = \lambda_k + \beta\gamma(Ax_{k+1} + By_{k+1} - b), \end{cases} \tag{1.7}$$

where $D_0 = D - (1-\tau)\beta B^\top B$ with an arbitrarily given positive-definite matrix $D \in \mathbb{R}^{n_2 \times n_2}$, and $\tau \in (0,1]$, $\gamma \in (0,1]$, and $\theta_k \geq 1$. By extrapolating a parameter v_k, the convergence rate of the APIPL-ADMM algorithm is proved. Since the ergodic average will destroy the sparsity and low-rank in sparse learning and low-rank learning [18, 20], in order to better apply the algorithm to practical problems, we analyze the non-ergodic convergence rate of APIPL-ADMM (1.7) so that it is possible to use in sparse domain and low-rank domain problems.

2. Preliminaries

In this section, we recall some basic definitions and well-known results that are useful for our further discussions.

Definition 2.1. ([21]) Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper convex function, and let $x \in \mathrm{dom}(f) := \{x : f(x) > -\infty\}$. A vector $\xi \in \mathbb{R}^n$ is called a subgradient of f at x if

$$f(y) \geq f(x) + \langle \xi, y - x \rangle, \quad \forall y \in \mathbb{R}^n.$$

The set of all subgradients of f at x is called the subdifferential of f at x and is denoted by $\partial f(x)$:

$$\partial f(x) = \{\xi \in \mathbb{R}^n : f(y) \geq f(x) + \langle \xi, y - x \rangle, \quad \forall y \in \mathbb{R}^n\}.$$

Definition 2.2. ([18]) Let $\{x_1, \ldots, x_K\}$ be a sequence produced by the algorithm. We say a convergence rate of the algorithm is nonergodic if it measures the optimality at x_K directly.

Lemma 2.1. *([20]) For any $x, y, z, \omega \in \mathbb{R}^n$, the following equations hold:*

$$2\langle x, y \rangle = \|x\|^2 + \|y\|^2 - \|x - y\|^2,$$
$$2\langle x - z, y - \omega \rangle = \|x - \omega\|^2 - \|z - \omega\|^2 - \|x - y\|^2 + \|z - y\|^2.$$

Lemma 2.2. *([20]) Let $f:\mathbb{R}^n\to\mathbb{R}$ be convex and (x^*,λ^*) be a KKT point of problem:*

$$\min_x \; f(x) \text{ s.t. } Ax=b.$$

Then

$$f(x)-f(x^*)+\langle\lambda^*,Ax-b\rangle\geq 0,\ \forall x.$$

Lemma 2.3. *([20]) Let $f:\mathbb{R}^n\to\mathbb{R}$ be convex and (x^*,λ^*) be a KKT point of problem:*

$$\min_x \; f(x) \text{ s.t. } Ax=b.$$

If $f(x)-f(x^*)+\langle\lambda^*,Ax-b\rangle\leq\alpha_1$ and $\|Ax-b\|\leq\alpha_2$, then

$$-\|\lambda^*\|\alpha_2\leq f(x)-f(x^*)\leq\|\lambda^*\|\alpha_2+\alpha_1.$$

3. Convergence Rate Analysis

In this section, we analyze the convergence rate of the APIPL-ADMM (1.7) with positive-indefinite matrix D_0 for solving problem (1.1).

For the sake of simplicity, we give several auxiliary variables:

$$\begin{aligned}
\overline{\lambda}_{k+1}&=\lambda_k+\frac{\beta}{\theta_k}(Ax_{k+1}+Bv_k-b),\\
\hat{\lambda}_k&=\lambda_k+\frac{\beta(1-\theta_k)}{\theta_k}(Ax_k+By_k-b),\\
z_{k+1}&=\frac{1}{\theta_k}y_{k+1}-\frac{1-\theta_k}{\theta_k}y_k,
\end{aligned}$$

and θ_k satisfies $\frac{1-\theta_{k+1}}{\theta_{k+1}}=\frac{1}{\theta_k}-\gamma$ and $\theta_{-1}=\frac{1}{\gamma}$.

Lemma 3.1. *For the definitions of $\overline{\lambda}_{k+1},\hat{\lambda}_k,\lambda_k,z_k,y_{k+1},v_k$, and θ_k, we have*

$$\begin{aligned}
&\hat{\lambda}_{k+1}-\hat{\lambda}_k=\frac{\beta}{\theta_k}[Ax_{k+1}+By_{k+1}-b-(1-\theta_k)(Ax_k+By_k-b)],\\
&\hat{\lambda}_{K+1}-\hat{\lambda}_0=\frac{\beta}{\theta_k}(Ax_{K+1}+By_{K+1}-b)+\beta\gamma\sum_{k=1}^{K}(Ax_k+By_k-b),\\
&\|\hat{\lambda}_{k+1}-\overline{\lambda}_{k+1}\|=\frac{\beta}{\theta_k}\|By_{k+1}-Bv_k\|,\\
&v_k-(1-\theta_k)y_k=\theta_k z_k.
\end{aligned}$$

Proof. By the definition of λ_{k+1} and $\frac{1-\theta_{k+1}}{\theta_{k+1}}=\frac{1}{\theta_k}-\gamma$, we have

$$\begin{aligned}
\hat{\lambda}_{k+1}&=\lambda_{k+1}+\frac{\beta(1-\theta_{k+1})}{\theta_{k+1}}(Ax_{k+1}+By_{k+1}-b)\\
&=\lambda_{k+1}+\beta\left(\frac{1}{\theta_k}-\gamma\right)(Ax_{k+1}+By_{k+1}-b)\\
&=\lambda_k+\beta\gamma(Ax_{k+1}+By_{k+1}-b)+\beta\left(\frac{1}{\theta_k}-\gamma\right)(Ax_{k+1}+By_{k+1}-b)\\
&=\lambda_k+\frac{\beta}{\theta_k}(Ax_{k+1}+By_{k+1}-b). \qquad (3.1)
\end{aligned}$$

Since $\hat{\lambda}_k = \lambda_k + \frac{\beta(1-\theta_k)}{\theta_k}(Ax_k + By_k - b)$, then

$$\lambda_k = \hat{\lambda}_k - \frac{\beta(1-\theta_k)}{\theta_k}(Ax_k + By_k - b)$$

and so,

$$\begin{aligned}\hat{\lambda}_{k+1} &= \hat{\lambda}_k - \frac{\beta(1-\theta_k)}{\theta_k}(Ax_k + By_k - b) + \frac{\beta}{\theta_k}(Ax_{k+1} + By_{k+1} - b) \\ &= \hat{\lambda}_k + \frac{\beta}{\theta_k}[Ax_{k+1} + By_{k+1} - b - (1-\theta_k)(Ax_k + By_k - b)].\end{aligned}$$

According to the definition of $\overline{\lambda}_{k+1}$, it follows from (3.1) that

$$\left\|\hat{\lambda}_{k+1} - \overline{\lambda}_{k+1}\right\| = \frac{\beta}{\theta_k}\|By_{k+1} - Bv_k\|.$$

By adding $\hat{\lambda}_{k+1} = \hat{\lambda}_k - \frac{\beta(1-\theta_k)}{\theta_k}(Ax_k + By_k - b) + \frac{\beta}{\theta_k}(Ax_{k+1} + By_{k+1} - b)$ from $k=0$ to $k=K$, we have

$$\begin{aligned}\hat{\lambda}_{K+1} - \hat{\lambda}_0 &= \sum_{k=0}^{K}(\hat{\lambda}_{k+1} - \hat{\lambda}_k) \\ &= \beta\sum_{k=0}^{K}\left[\frac{1}{\theta_k}(Ax_{k+1} + By_{k+1} - b) - \frac{1-\theta_k}{\theta_k}(Ax_k + By_k - b)\right] \\ &= \beta\sum_{k=0}^{K}\left[\frac{1}{\theta_k}(Ax_{k+1} + By_{k+1} - b) - \frac{1}{\theta_{k-1}}(Ax_k + By_k - b) + \gamma(Ax_k + By_k - b)\right] \\ &= \frac{\beta}{\theta_K}(Ax_{K+1} + By_{K+1} - b) + \beta\gamma\sum_{k=1}^{K}(Ax_k + By_k - b).\end{aligned}$$

Note that

$$\begin{aligned}(1-\theta_k)y_k + \theta_k z_k &= (1-\theta_k)y_k + \frac{\theta_k}{\theta_{k-1}}[y_k - (1-\theta_{k-1})y_{k-1}] \\ &= y_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(y_k - y_{k-1}) \\ &= v_k.\end{aligned}$$

Therefore, $v_k - (1-\theta_k)y_k = \theta_k z_k$. □

The following lemma plays a key role in the analysis of convergence rate of APIPL-ADMM (1.7) in the sense of objective values.

Lemma 3.2. *For the iterative sequence (x_{k+1}, y_{k+1}) generated by Algorithm (1.7), we have*

$$\begin{aligned}&f(x_{k+1}) + g(y_{k+1}) - f(x) - g(y) \\ &\leq -\left\langle\overline{\lambda}_{k+1}, Ax_{k+1} + By_{k+1} - Ax - By\right\rangle - \frac{\beta}{\theta_k}\langle By_{k+1} - Bv_k, By_{k+1} - By\rangle \\ &\quad - \langle D_0(y_{k+1} - y_k), y_{k+1} - y\rangle.\end{aligned} \tag{3.2}$$

Proof. Let

$$\hat{\nabla} f(x_{k+1}) := -A^\top \lambda_k - \frac{\beta}{\theta_k} A^\top (Ax_{k+1} + Bv_k - b) = -A^\top \overline{\lambda}_{k+1}$$

and

$$\begin{aligned}\hat{\nabla} g(y_{k+1}) &:= -B^\top \lambda_k - \frac{\beta}{\theta_k} B^\top (Ax_{k+1} + By_{k+1} - b) - D_0(y_{k+1} - y_k) \\ &= -B^\top \overline{\lambda}_{k+1} - \frac{\beta}{\theta_k} B^\top B(y_{k+1} - v_k) - D_0(y_{k+1} - y_k).\end{aligned}$$

For Algorithm (1.7), we have $\hat{\nabla} f(x_{k+1}) \in \partial f(x_{k+1})$ and $\hat{\nabla} g(y_{k+1}) \in \partial g(y_{k+1})$. By the convexity of f and g, we have

$$\begin{aligned}f(x_{k+1}) - f(x) &\le \left\langle \hat{\nabla} f(x_{k+1}), x_{k+1} - x \right\rangle \\ &= -\left\langle \overline{\lambda}_{k+1}, Ax_{k+1} - Ax \right\rangle,\end{aligned} \tag{3.3}$$

and

$$\begin{aligned}g(y_{k+1}) - g(y) &\le \left\langle \hat{\nabla} g(y_{k+1}), y_{k+1} - y \right\rangle \\ &= -\left\langle \overline{\lambda}_{k+1}, By_{k+1} - By \right\rangle - \frac{\beta}{\theta_k} \langle By_{k+1} - Bv_k, By_{k+1} - By \rangle \\ &\quad - \langle D_0 y_{k+1} - D_0 y_k, y_{k+1} - y \rangle .\end{aligned} \tag{3.4}$$

Thus the desired conclusion is obtained by adding (3.3) and (3.4). □

Lemma 3.3. *Let (x^*, y^*) be a solution to problem (1.1). For the iterative sequence (x_k, y_k) generated by Algorithm (1.7),*

$$\begin{aligned}&f(x_{K+1}) + g(y_{K+1}) - f(x^*) - g(y^*) + \langle \lambda^*, Ax_{K+1} + By_{K+1} - b \rangle \\ &\le \theta_K \left(\frac{1}{2\beta} \left\| \hat{\lambda}_0 - \lambda^* \right\|^2 + \frac{\beta}{2} \| Bz_0 - By^* \|^2 + \frac{1}{2} \| D_0 y_0 - y^* \|^2 \right),\end{aligned} \tag{3.5}$$

and

$$\begin{aligned}&\left\| \frac{1}{\theta_K}(Ax_{K+1} + By_{K+1} - b) + \gamma \sum_{k=1}^{K} (Ax_k + By_k - b) \right\| \\ &\le \frac{2}{\beta} \left\| \hat{\lambda}_0 - \lambda^* \right\| + \| Bz_0 - By^* \| + \sqrt{\beta} \, \| D_0 y_0 - y^* \| .\end{aligned} \tag{3.6}$$

Proof. Taking $(x, y) = (x^*, y^*)$ and $(x, y) = (x_k, y_k)$ in (3.2), respectively, we obtain two inequalities

$$\begin{aligned}&f(x_{k+1}) + g(y_{k+1}) - f(x^*) - g(y^*) \\ &\le -\left\langle \overline{\lambda}_{k+1}, Ax_{k+1} + By_{k+1} - Ax^* - By^* \right\rangle - \frac{\beta}{\theta_k} \langle By_{k+1} - Bv_k, By_{k+1} - By^* \rangle \\ &\quad - \langle D_0(y_{k+1} - y_k), y_{k+1} - y^* \rangle ,\end{aligned} \tag{3.7}$$

and

$$
\begin{aligned}
&f(x_{k+1})+g(y_{k+1})-f(x_k)-g(y_k)\\
&\le -\left\langle \overline{\lambda}_{k+1}, Ax_{k+1}+By_{k+1}-Ax_k-By_k\right\rangle-\frac{\beta}{\theta_k}\langle By_{k+1}-Bv_k, By_{k+1}-By_k\rangle\\
&\quad-\langle D_0(y_{k+1}-y_k), y_{k+1}-y_k\rangle .
\end{aligned}
\tag{3.8}
$$

Multiplying the inequality (3.7) by θ_k and multiplying the second inequality (3.8) by $1-\theta_k$, summing them together, and using $Ax^*+By^*=b$, we have

$$
\begin{aligned}
&f(x_{k+1})+g(y_{k+1})-(1-\theta_k)(f(x_k)+g(y_k))-\theta_k(f(x^*)+g(y^*))\\
&\le -\left\langle \overline{\lambda}_{k+1}, Ax_{k+1}+By_{k+1}-b-(1-\theta_k)(Ax_k+By_k-b)\right\rangle\\
&\quad-\frac{\beta}{\theta_K}\langle By_{k+1}-Bv_k, By_{k+1}-(1-\theta_k)By_k-\theta_k By^*\rangle\\
&\quad-\langle D_0y_{k+1}-D_0y_k, y_{k+1}-y_k\rangle-\langle D_0y_{k+1}-D_0y_k, \theta_k(y_k-y^*)\rangle .
\end{aligned}
$$

Dividing both sides by θ_k, adding $\left\langle \lambda^*, \frac{1}{\theta_k}(Ax_{k+1}+By_{k+1}-b)-\frac{1-\theta_k}{\theta_k}(Ax_k+By_k-b)\right\rangle$ to both sides, together with the positive-indefiniteness of D_0, $Ax-Ax^*=Ax-b+By^*$, and Lemmas 2.1 and 3.1, we have

$$
\begin{aligned}
&\frac{f(x_{k+1})+g(y_{k+1})-f(x^*)-g(y^*)+\langle\lambda^*, Ax_{k+1}+By_{k+1}-b\rangle}{\theta_k}\\
&-\frac{1-\theta_k}{\theta_k}(f(x_k)+g(y_k)-f(x^*)-g(y^*)+\langle\lambda^*, Ax_k+By_k-b\rangle)\\
&\le -\frac{1}{\beta}\left\langle \overline{\lambda}_{k+1}-\lambda^*, \hat{\lambda}_{k+1}-\hat{\lambda}_k\right\rangle-\frac{\beta}{\theta_k^2}\langle By_{k+1}-Bv_k, By_{k+1}-(1-\theta_k)By_k-\theta_k By^*\rangle\\
&\quad-\frac{1}{\theta_k}\langle D_0y_{k+1}-D_0y_k, y_{k+1}-y_k\rangle-\langle D_0y_{k+1}-D_0y_k, y_k-y^*\rangle\\
&=\frac{1}{2\beta}\left(\|\hat{\lambda}_k-\lambda^*\|^2-\|\hat{\lambda}_{k+1}-\lambda^*\|^2-\|\hat{\lambda}_k-\overline{\lambda}_{k+1}\|^2+\|\hat{\lambda}_{k+1}-\overline{\lambda}_{k+1}\|^2\right)\\
&\quad+\frac{\beta}{2\theta_k^2}\left(\|Bv_k-(1-\theta_k)By_k-\theta_k By^*\|^2-\|By_{k+1}-(1-\theta_k)By_k-\theta_k By^*\|^2-\|By_{k+1}-Bv_k\|^2\right)\\
&\quad+\frac{1}{2\theta_k}\left(\|D_0y_{k+1}-y_{k+1}\|^2+\|D_0y_k-y_k\|^2-\|D_0y_{k+1}-y_k\|^2-\|D_0y_k-y_{k+1}\|^2\right)\\
&\quad+\frac{1}{2}\left(\|D_0y_k-y^*\|^2+\|D_0y_{k+1}-y_k\|^2-\|D_0y_{k+1}-y^*\|^2-\|D_0y_k-y_k\|^2\right)\\
&\le\frac{1}{2\beta}\left(\|\hat{\lambda}_k-\lambda^*\|^2-\|\hat{\lambda}_{k+1}-\lambda^*\|^2\right)+\frac{\beta}{2}\left(\|Bz_k-By^*\|^2-\|Bz_{k+1}-By^*\|^2\right)\\
&\quad+\frac{1}{2}\left(\|D_0y_k-y^*\|^2-\|D_0y_{k+1}-y^*\|^2\right).
\end{aligned}
$$

Using $\frac{1-\theta_k}{\theta_k} = \frac{1}{\theta_{k-1}} - \gamma$ and $\theta_{-1} = \frac{1}{\gamma}$, and summing from $k = 0$ to $k = K$, we have

$$\begin{aligned}
&\frac{f(x_{K+1}) + g(y_{K+1}) - f(x^*) - g(y^*) + \langle \lambda^*, Ax_{K+1} + By_{K+1} - b\rangle}{\theta_K} \\
&+ \gamma \sum_{k=1}^{K} (f(x_k) + g(y_k) - f(x^*) - g(y^*) + \langle \lambda^*, Ax_k + By_k - b\rangle) \\
&\leq \frac{1}{2\beta}\left(\|\hat{\lambda}_0 - \lambda^*\|^2 - \|\hat{\lambda}_{K+1} - \lambda^*\|^2\right) + \frac{\beta}{2}(\|Bz_0 - By^*\|^2 - \|Bz_{k+1} - By^*\|^2) \\
&\quad + \frac{1}{2}(\|D_0 y_0 - y^*\|^2 - \|D_0 y_{K+1} - y^*\|^2) \\
&\leq \frac{1}{2\beta}\left(\|\hat{\lambda}_0 - \lambda^*\|^2 - \|\hat{\lambda}_{K+1} - \lambda^*\|^2\right) + \frac{\beta}{2}\|Bz_0 - By^*\|^2 + \frac{1}{2}\|D_0 y_0 - y^*\|^2,
\end{aligned}$$

i.e.,

$$\begin{aligned}
&\frac{f(x_{K+1}) + g(y_{K+1}) - f(x^*) - g(y^*) + \langle \lambda^*, Ax_{K+1} + By_{K+1} - b\rangle}{\theta_K} \\
&\leq \frac{1}{2\beta}\left(\|\hat{\lambda}_0 - \lambda^*\|^2 - \|\hat{\lambda}_{K+1} - \lambda^*\|^2\right) + \frac{\beta}{2}\|Bz_0 - By^*\|^2 + \frac{1}{2}\|D_0 y_0 - y^*\|^2.
\end{aligned}$$

From Lemma 2.2, we deduce that (3.5) and

$$\begin{aligned}
\|\hat{\lambda}_{K+1} - \hat{\lambda}_0\| - \|\hat{\lambda}_0 - \lambda^*\| &\leq \|\hat{\lambda}_{K+1} - \lambda^*\| \\
&\leq \sqrt{\|\hat{\lambda}_0 - \lambda^*\|^2 + \beta^2\|Bz_0 - By^*\|^2 + \beta\|D_0 y_0 - y^*\|^2} \\
&\leq \|\hat{\lambda}_0 - \lambda^*\| + \beta\|Bz_0 - By^*\| + \sqrt{\beta}\|D_0 y_0 - y^*\|,
\end{aligned}$$

which leads to

$$\|\hat{\lambda}_{K+1} - \hat{\lambda}_0\| \leq 2|\hat{\lambda}_0 - \lambda^*\| + \beta\|Bz_0 - By^*\| + \sqrt{\beta}\|D_0 y_0 - y^*\|.$$

It follows from Lemma 3.1 that

$$\begin{aligned}
&\left\|\frac{1}{\theta_K}(Ax_{K+1} + By_{K+1} - b) + \gamma\sum_{k=1}^{K}(Ax_k + By_k - b)\right\| \\
&= \left\|\frac{1}{\beta}\hat{\lambda}_{K+1} - \hat{\lambda}_0\right\| \\
&\leq \frac{2}{\beta}\left\|\hat{\lambda}_0 - \lambda^*\right\| + \|Bz_0 - By^*\| + \sqrt{\beta}\,\|D_0 y_0 - y^*\|.
\end{aligned}$$

i.e.,

$$\begin{aligned}
&\left\|\frac{1}{\theta_K}(Ax_{K+1} + By_{K+1} - b) + \gamma\sum_{k=1}^{K}(Ax_k + By_k - b)\right\| \\
&\leq \frac{2}{\beta}\left\|\hat{\lambda}_0 - \lambda^*\right\| + \|Bz_0 - By^*\| + \sqrt{\beta}\,\|D_0 y_0 - y^*\|.
\end{aligned}$$

It therefore yields that (3.5) and (3.6) hold. □

In order to obtain an estimate of the bound from above of the violation for constraint in the form of $\|Ax + By - b\|$ instead of (3.6), we recall the following auxiliary result.

Lemma 3.4. [20, Lemma 3.18] *Consider a sequence $(c_k)_{k=1}^{\infty}$ of vectors, if (c_k) satisfies*

$$\left\| \left[\frac{1}{\gamma} + K\left(\frac{1}{\gamma} - 1\right)\right] c_{K+1} + \sum_{k=1}^{K} c_k \right\| \leq c, \ \forall\, K = 0,1,2\cdots$$

where $0 < \gamma < 1$. Then $\left\|\sum_{k=1}^{K} c_k\right\| < c$ for all $K = 1,2,\cdots$.

We now prove the main result on the convergence rate in the nonergodic sense of Algorithm (1.7) in terms of objective values as well as the bound above of the violation of constraint.

Theorem 3.1. *Assume that (x^*, y^*) is a solution of problem (1.1). For Algorithm (1.7), we have*

$$0 \leq f(x_{K+1}) + g(y_{K+1}) - f(x^*) - g(y^*) \leq \frac{2\Gamma\|\lambda^*\|}{1+K(1-\gamma)} + \frac{\Lambda}{1+K(1-\gamma)},$$

and

$$\|Ax_{K+1} + By_{K+1} - b\| \leq \frac{2\Gamma}{1+K(1-\gamma)},$$

where

$$\Gamma = \frac{2}{\beta}\left\|\hat{\lambda}_0 - x^*\right\| + \|Bz_0 - By^*\| + \sqrt{\beta}\,\|D_0 y_0 - y^*\|, \tag{3.9}$$

$$\Lambda = \frac{1}{2\beta}\left\|\hat{\lambda}_0 - \lambda^*\right\| + \frac{\beta}{2}\|Bz_0 - By^*\| + \frac{1}{2}\|D_0 y_0 - y^*\|. \tag{3.10}$$

Proof. Note that $\frac{1-\theta_{k+1}}{\theta_{k+1}} = \frac{1}{\theta_k} - \gamma$, and $\theta_{-1} = \frac{1}{\gamma}$. Then $\theta_0 = 1$ and

$$\frac{1}{\theta_k} = \frac{1}{\theta_{k-1}} + 1 - \gamma = \frac{1}{\theta_0} + k(1-\gamma) = 1 + k(1-\gamma).$$

It implies that $\theta_k = \frac{1}{1+k(1-\gamma)}$. Set $c_k := Ax_k + By_k - b$. Using (3.6) yields

$$\begin{aligned}
&\left\| \left[\frac{1}{\gamma} + K\left(\frac{1}{\gamma} - 1\right)\right] c_{K+1} + \sum_{k=1}^{K} c_k \right\| \\
&= \left\| \frac{1}{\gamma\theta_k}(Ax_{K+1} + By_{K+1} - b) + \sum_{k=1}^{K}(Ax_k + By_k - b) \right\| \\
&\leq \frac{1}{\gamma}\left(\frac{2}{\beta}\left\|\hat{\lambda}_0 - \hat{\lambda}^*\right\| + \|Bz_0 - By^*\| + \sqrt{\beta}\,\|D_0 y_0 - y^*\| \right) \\
&= \frac{\Gamma}{\gamma}, \ \forall K = 0,1,2,\cdots,
\end{aligned} \tag{3.11}$$

where $\Gamma = \frac{2}{\beta}\left\|\hat{\lambda}_0 - x^*\right\| + \|Bz_0 - By^*\| + \sqrt{\beta}\,\|D_0 y_0 - y^*\|$. It therefore follows from Lemma 3.4 that

$$\left\| \sum_{k=1}^{K} c_k \right\| \leq \frac{\Gamma}{\gamma}, \ \forall K = 1,2,\cdots.$$

From (3.11), we conclude that

$$\begin{aligned}
&\left\|\left[\frac{1}{\gamma}+K\left(\frac{1}{\gamma}-1\right)\right]c_{K+1}\right\|-\left\|\sum_{k=1}^{K}c_k\right\| \\
\leq &\left\|\left[\frac{1}{\gamma}+K\left(\frac{1}{\gamma}-1\right)\right]c_{K+1}+\sum_{k=1}^{K}c_k\right\| \leq \frac{\Gamma}{\gamma},\ \forall K=0,1,2,\cdots,
\end{aligned}$$

and so,

$$\left\|\left[\frac{1}{\gamma}+K\left(\frac{1}{\gamma}-1\right)\right]c_{K+1}\right\| \leq \left\|\sum_{k=1}^{K}c_k\right\|+\frac{\Gamma}{\gamma}=\frac{2\Gamma}{\gamma},\ \forall K=0,1,2,\cdots.$$

Consequently, one has $\|c_{K+1}\| \leq \frac{2\Gamma}{1+K(1-\gamma)}$ for all $K=0,1,2,\cdots$. Namely, we have

$$\|Ax_{K+1}+By_{K+1}-b\| \leq \frac{2\Gamma}{1+K(1-\gamma)},$$

which together with (3.5) yields that

$$\begin{aligned}
&f(x_{K+1})+g(y_{K+1})-f(x^*)-g(y^*)+\langle\lambda^*, Ax_{K+1}+By_{K+1}-b\rangle \\
&\leq \theta_K\left(\frac{1}{2\beta}\left\|\hat{\lambda}_0-\lambda^*\right\|^2+\frac{\beta}{2}\|Bz_0-By^*\|^2+\frac{1}{2}\|D_0y_0-y^*\|^2\right) \\
&=\theta_K\Lambda \\
&=\frac{\Lambda}{1+K(1-\gamma)},
\end{aligned}$$

where the last equality is obtained by $\theta_k=\frac{1}{1+k(1-\gamma)}$. Consequently, by Lemma 2.3, we obtain

$$-\frac{2\Gamma\|\lambda^*\|}{1+K(1-\gamma)} \leq f(x_{K+1})+g(y_{K+1})-f(x^*)-g(y^*) \leq \frac{2\Gamma\|\lambda^*\|}{1+K(1-\gamma)}+\frac{\Lambda}{1+K(1-\gamma)},$$

for all $K=0,1,2,\cdots$. Since (x^*,y^*) is a solution of problem (1.1) and $\gamma\in(0,1]$, $-\frac{2\Gamma\|\lambda^*\|}{1+K(1-\gamma)} \leq 0$ and $f(x_{K+1})+g(y_{K+1})-f(x^*)-g(y^*) \geq 0$ for all $K=0,1,2,\cdots$. So,

$$0 \leq f(x_{K+1})+g(y_{K+1})-f(x^*)-g(y^*) \leq \frac{2\Gamma\|\lambda^*\|}{1+K(1-\gamma)}+\frac{\Lambda}{1+K(1-\gamma)},$$

for all $K=0,1,2,\cdots$. □

The following result shows the convergence rate in the nonergodic sense of Algorithm (1.7) in terms of iterative sequence.

Theorem 3.2. *Assume that (x^*,y^*) is a solution of problem (1.1), f is strongly convex with constant $\iota_1>0$, and g is strongly convex with constant $\iota_2>0$. For Algorithm (1.7) with $\gamma\in(0,1)$, we have*

$$\|(x_K,y_K)-(x^*,y^*)\| \leq \Theta\frac{1}{\sqrt{K}},$$

where $\Theta=\sqrt{\frac{2\Gamma\|\lambda^*\|+\Lambda}{1-\gamma+\frac{\gamma}{K}}}\frac{\sqrt{2}}{\sqrt{\min\{\iota_1,\iota_2\}}}$, Γ is defined as (3.9), and Λ is defined as (3.10).

Proof. It follows from Theorem 3.1 that

$$0 \le f(x_K) + g(y_K) - f(x^*) - g(y^*) \le \frac{2\Gamma\|\lambda^*\| + \Lambda}{(1-\gamma)K + \gamma}.$$

Since f is strongly convex with constant $\iota_1 > 0$ and g is strongly convex with constant $\iota_2 > 0$, then

$$f(x_K) + g(y_K) - f(x^*) - g(y^*) \ge \frac{\iota_1}{2}\|x_K - x^*\|^2 + \frac{\iota_2}{2}\|y_K - y^*\|^2 \ge \iota\|(x_K, y_K) - (x^*, y^*)\|^2,$$

where $\iota = \min\{\frac{\iota_1}{2}, \frac{\iota_2}{2}\}$. Therefore, one has

$$\|(x_K, y_K) - (x^*, y^*)\| \le \sqrt{\frac{2\Gamma\|\lambda^*\| + \Lambda}{\iota(1-\gamma)K + \gamma\iota}} \le \sqrt{\frac{1}{K}}\sqrt{\frac{2\Gamma\|\lambda^*\| + \Lambda}{1-\gamma+\frac{\gamma}{K}}}\frac{\sqrt{2}}{\sqrt{\min\{\iota_1, \iota_2\}}},$$

i.e., $\|(x_K, y_K) - (x^*, y^*)\| \le \Theta\frac{1}{\sqrt{K}}$. □

4. Numerical Experiments

In this section, we demonstrate the computational performance of APIPL-ADMM with positive-indefinite matrix D_0 by the well-known LASSO model and compare APIPL-ADMM with PIPL-ADMM and the existing PD-ADMM. To this end, taking the arbitrarily positive-definite matrix $D := \tau(\mu I_{n_2} - \beta B^\top B)$ with $\mu > \|\beta B^\top B\|$. Then one obtains the positive-indefinite matrix $D_0 := \tau\mu I_{n_2} - \beta B^\top B$. All codes are written in Matlab and all experiments are performed in Matlab R2015b on a workstation with an Intel(R) Core(TM) i7-8550U CPU (1.80 GHz) and 8GB RAM.

The LASSO model is given as

$$\min_y \frac{1}{2}\|Ay - b\|^2 + \sigma\|y\|_1, \tag{4.1}$$

where $\|y\|_1 := \Sigma_{i=1}^n |y_i|$, $A \in \mathbb{R}^{m\times n}$ is a design matrix usually with $m \ll n$, m is the number of date point, n is the number of features, $b \in \mathbb{R}^m$ is the response vector, and $\sigma > 0$ is a regularization parameter.

By a new auxiliary variable x, (4.1) can be rewritten as the form:

$$\min\left\{\frac{1}{2}\|x - b\|^2 + \sigma\|y\|_1 \,|\, x - Ay = 0, x \in \mathbb{R}^m, y \in \mathbb{R}^n\right\}, \tag{4.2}$$

The augmented Lagrangian function of problem (4.2) is defined by

$$\mathscr{L}_\beta(x, y, \lambda) := \frac{1}{2}\|x - b\|^2 + \sigma\|y\|_1 - \lambda^\top(x - Ay) + \frac{\beta}{2}\|x - Ay\|^2,$$

where $\lambda \in \mathbb{R}^m$ denotes the Lagrangian multiplier and $\beta > 0$ is a penalty parameter. Then apply the APIPL-ADMM algorithm to (4.2), we have

$$x_{k+1} = \arg\min\left\{\frac{1}{2}\|x - b\|^2 - (\lambda_k)^\top(x - Av_k) + \frac{\beta}{2\theta_k}\|x - Av_k\|^2 \,|\, x \in \mathbb{R}^m\right\}, \tag{4.3}$$

$$y_{k+1} = \arg\min\left\{\sigma\|y\|_1 + \frac{\tau\mu}{2}\|y - (y_k + \frac{1}{\tau\mu}q_k)\|^2 \,|\, y \in \mathbb{R}^n\right\} \tag{4.4}$$

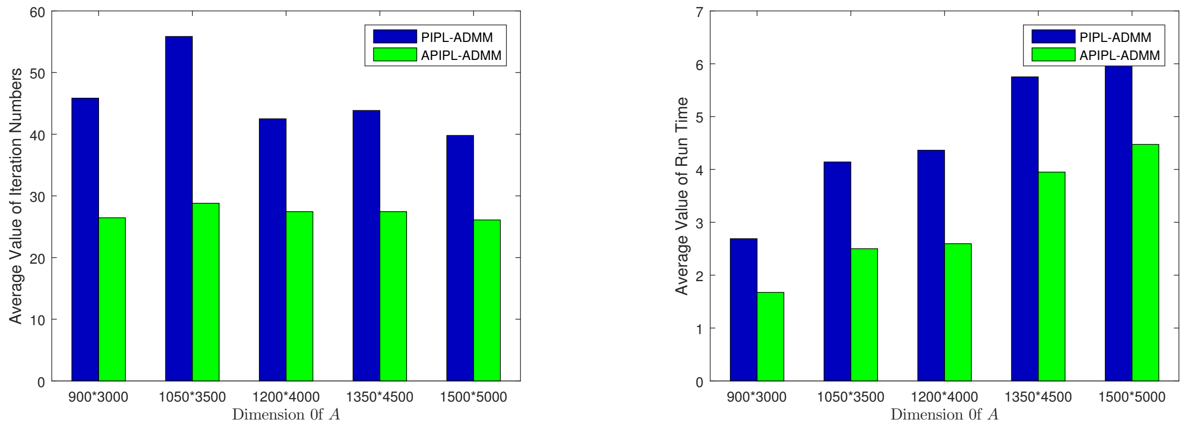

FIGURE 1. The average number of iterations and average runtime of APIPL-ADMM and PIPL-ADMM in different dimensions

and

$$\lambda_{k+1} = \lambda_k - \gamma\beta(x_{k+1} - Ay_{k+1}), \tag{4.5}$$

where $q_k := -A^\top(\lambda_k - \frac{\beta}{\theta_k}(x_{k+1} - Ay_k))$. Then the solutions of the subproblems (4.3) and (4.4) are given respectively by the following explicit form:

$$x_{k+1} = \frac{\theta_k}{\theta_k + \beta}(b + \lambda_k + \beta A v_k).$$

and

$$y_{k+1} = S_{\sigma/\tau\mu}(y_k + \frac{q_k}{\tau\mu}),$$

where $S_\delta(t)$ is the soft thresholding operator [17] defined as

$$(S_\delta(t))_i := (1 - \delta/|t_i|)_+ \cdot t_i, i = 1, 2, \cdots, m.$$

We generate the data by the same method in [17]: we choose $A_{ij} \sim \mathcal{N}(0,1)$ and then scaled the columns to have unit norm. We use the function 'sprandn' to generate a sparse vector y^* which has approximately density $= 100/n$ non-zeros entries taken from the normal distribution with zero mean and unit variance. We generate b via $b := Ay^* + e$, where e is a small white noise taken from $e \sim \mathcal{N}(0, 10^{-3}I)$. We test five cases of dimension of A ranging from 900×300 to 1500×5000, and set $\mu := \beta\|A^\top A\|$. We set the regularization parameter σ to 0.1 and the ADMM parameter β to $\frac{2-\gamma}{\gamma|\gamma-1|}$. Since the small proximal term can allow for the larger step size, we take $\tau := \frac{|\gamma-1|}{5\beta\gamma} + \frac{4}{5}$. We pick up $\gamma := 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75$. The termination criteria are defined by $\|x_{k+1} - Ay_{k+1}\| \leq \varepsilon^{pri}$ and $\|\beta A(y_{k+1} - y_k)\| \leq \varepsilon^{dual}$, where $\varepsilon^{pri} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel}\max\{\|x_{k+1}\|, \|Ay_{k+1}\|\}$, and $\varepsilon^{dual} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel}\|y_{k+1}\|$ with ε^{abs} and ε^{rel} setting respectively to be 10^{-4} and 10^{-2}. We choose the initial point in two different ways, the first is $(x_0, y_0, \lambda_0) := (0, 0, 0)$, and the second is $(x_0, y_0, \lambda_0) := (0, \hat{y}_0, 0)$ where $\hat{y}_0$ is randomly generated. Then the average number of iterations and average running time are taken into account.

TABLE 1. The average number of iterations and runtime of the APIPL-ADMM and PIPL-ADMM

γ	$m \times n$	PIPL-ADMM Iter.1	Time.1	APIPL-ADMM Iter.2	Time.2	$\frac{Iter.2}{Iter.1}$	$\frac{Time2}{Time1}$
	900×3000	45.5	2.72	38	2.43	0.83	0.89
	1050×3500	50	4.01	42	3.34	0.83	0.83
$\gamma = 0.3$	1200×4000	42.5	4.51	38	4.05	0.89	0.89
	1350×4500	44.5	5.81	39.5	6.32	0.88	0.88
	1500×5000	38.5	6.20	35.5	5.83	0.92	0.94
	900×3000	45.5	2.65	34.5	2.46	0.75	0.78
	1050×3500	51.5	4.50	37.5	4.26	0.72	0.71
$\gamma = 0.35$	1200×4000	42.5	4.44	34	4.34	0.80	0.80
	1350×4500	45	5.89	35	5.69	0.78	0.79
	1500×5000	39	6.39	32.5	6.64	0.83	0.82
	900×3000	46.5	2.76	30.5	2.46	0.65	0.67
	1050×3500	51.5	4.22	33.5	3.49	0.64	0.64
$\gamma = 0.4$	1200×4000	43	4.87	31.5	4.07	0.73	0.76
	1350×4500	45.5	5.95	32.5	5.36	0.71	0.72
	1500×5000	39.5	6.48	30	6.10	0.75	0.75
	900×3000	46.5	2.76	27.5	1.68	0.58	0.60
	1050×3500	52.5	4.35	30	2.43	0.56	0.55
$\gamma = 0.45$	1200×4000	43.5	4.52	29	3.03	0.66	0.66
	1350×4500	46	6.08	29	3.97	0.62	0.64
	1500×5000	39.5	6.47	28	4.61	0.70	0.70
	900×3000	47.5	2.95	25.5	1.63	0.53	0.54
	1050×3500	52.5	4.33	27.5	2.27	0.52	0.51
$\gamma = 0.5$	1200×4000	44	4.63	27	2.87	0.61	0.61
	1350×4500	46.5	6.33	27.5	3.71	0.57	0.58
	1500×5000	40.5	6.79	25.5	4.29	0.62	0.63
	900×3000	48	2.99	23.5	1.39	0.48	0.46
	1050×3500	54	4.63	26	2.22	0.47	0.47
$\gamma = 0.55$	1200×4000	45	4.92	24.5	2.61	0.54	0.53
	1350×4500	47	6.33	24.5	3.28	0.51	0.51
	1500×5000	41	6.80	24	3.96	0.63	0.57

In order to exhibit the efficiency and advantages of APIPL-ADMM, we plot the results in terms of iterations in Figure 1 and Figure 2. Tables 1 and 2 list the average number of iterations and runtime in seconds of the APIPL-ADMM and PIPL-ADMM schemes applied to problem (4.2) for different dimensions of A and parameter γ. Tables 3 and 4 list the average number of iterations and runtime in seconds of the APIPL-ADMM and PD-ADMM schemes applied to problem (4.2) for different dimensions of A and parameter γ. From the numerical experimental results, it can be seen that the APIPL-ADMM performs much better than PIPL-ADMM and PD-ADMM both in the number of iterations and runtime.

TABLE 2. The average number of iterations and runtime of the APIPL-ADMM and PIPL-ADMM

		PIPL-ADMM		APIPL-ADMM			
γ	$m \times n$	Iter.1	Time.1	Iter.2	Time.2	$\frac{Iter.2}{Iter.1}$	$\frac{Time2}{Time1}$
$\gamma = 0.6$	900×3000	43	2.41	22.5	1.25	0.53	0.53
	1050×3500	47	3.63	24	1.80	0.51	0.51
	1200×4000	39.5	3.98	24	2.39	0.61	0.61
	1350×4500	39.5	5.12	23	2.91	0.59	0.59
	1500×5000	41	5.71	22.5	3.46	0.62	0.61
$\gamma = 0.65$	900×3000	44	2.52	22	1.21	0.50	0.49
	1050×3500	48	3.91	22	1.73	0.46	0.45
	1200×4000	40.5	4.19	22	2.18	0.55	0.53
	1350×4500	40.5	5.23	21.5	2.80	0.54	0.55
	1500×5000	37	5.82	21.5	3.38	0.58	0.58
$\gamma = 0.7$	900×3000	45	2.59	20.5	1.13	0.46	0.44
	1050×3500	49	3.85	22	1.64	0.45	0.43
	1200×4000	41.5	4.16	22	2.16	0.53	0.53
	1350×4500	41.5	5.33	21.5	2.75	0.53	0.53
	1500×5000	43	6.04	20.5	3.18	0.54	0.53
$\gamma = 0.75$	900×3000	47	2.59	20	1.12	0.43	31.63
	1050×3500	51	4.03	23.5	1.85	0.46	0.46
	1200×4000	43	4.37	22.5	2.26	0.53	0.52
	1350×4500	42.5	5.48	20.5	2.73	0.49	0.51
	1500×5000	39	6.21	21	3.30	0.54	0.53

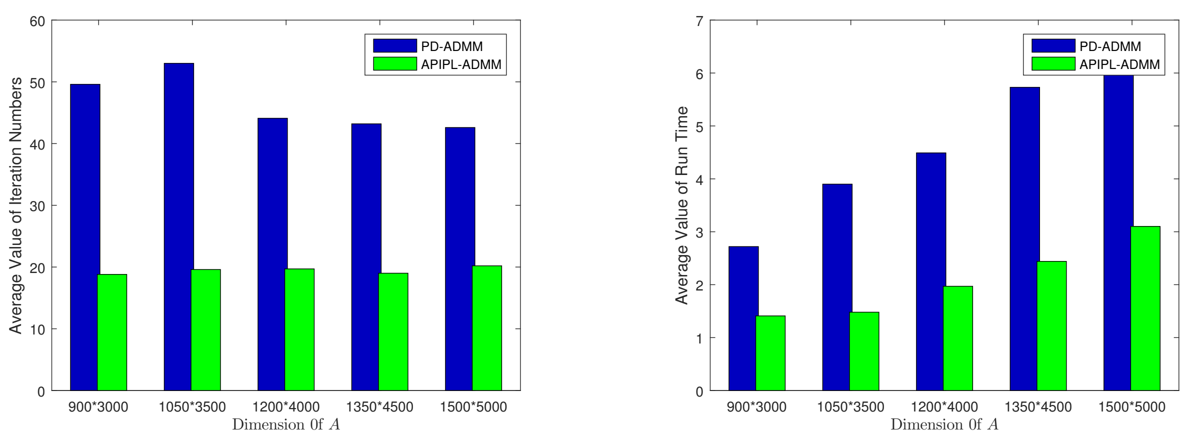

FIGURE 2. The average number of iterations and average runtime of APIPL-ADMM and PD-ADMM in different dimensions

5. CONCLUSIONS

An APIPL-ADMM algorithm with positive-indefinite proximal matrix was proposed for solving the two-block linearly constrained separable convex optimization problem. It was

TABLE 3. The average number of iterations and runtime of the APIPL-ADMM and PD-ADMM

		PIPL-ADMM		APIPL-ADMM			
γ	$m \times n$	Iter.1	Time.1	Iter.2	Time.2	$\frac{Iter.2}{Iter.1}$	$\frac{Time2}{Time1}$
	900×3000	46	2.55	31	1.68	0.67	0.66
	1050×3500	49	3.59	32	2.57	0.65	0.71
	1200×4000	41	4.02	30	2.89	0.73	0.72
	1350×4500	40	5.20	29	3.63	0.72	0.70
	1500×5000	40	6.05	30	4.47	0.75	0.74
	900×3000	46	2.45	27	1.48	0.59	0.61
	1050×3500	50	3.64	28	2.05	0.56	0.56
$\gamma = 0.35$	1200×4000	41	3.98	27	2.63	0.66	0.66
	1350×4500	41	5.15	26	3.48	0.63	0.68
	1500×5000	40	6.18	26	4.00	0.65	0.65
	900×3000	47	2.57	24	1.28	0.51	0.50
	1050×3500	50	3.60	24	1.87	0.48	0.52
$\gamma = 0.4$	1200×4000	42	4.10	24	2.48	0.57	0.61
	1350×4500	41	5.10	23	2.82	0.56	0.55
	1500×5000	40	6.16	24	3.68	0.60	0.60
	900×3000	48	2.62	20	1.14	0.42	0.44
	1050×3500	51	3.87	21	1.55	0.41	0.40
$\gamma = 0.45$	1200×4000	43	4.19	22	2.15	0.51	0.51
	1350×4500	42	5.58	21	2.62	0.50	0.47
	1500×5000	41	6.32	22	3.39	0.54	0.54
	900×3000	49	2.81	18	1.01	0.37	0.36
	1050×3500	52	3.72	18	1.29	0.35	0.35
$\gamma = 0.5$	1200×4000	43	4.19	19	1.79	0.44	0.43
	1350×4500	42	5.27	19	2.32	0.45	0.44
	1500×5000	42	6.46	20	2.98	0.48	0.46
	900×3000	49	2.65	16	0.86	0.33	0.32
	1050×3500	53	3.85	17	1.22	0.32	0.32
$\gamma = 0.55$	1200×4000	44	4.35	18	1.76	0.41	0.40
	1350×4500	43	5.43	17	2.12	0.40	0.39
	1500×5000	43	6.68	18	2.80	0.42	0.42

proved that the non-ergodic convergence rate of the APIPL-ADMM in the sense of objective values is $O(\frac{1}{K})$ and the non-ergodic convergence rates of the APIPL-ADMM in the sense of iterative sequence are $O(\frac{1}{\sqrt{K}})$, where K is the number of iterations. The effectiveness of the acceleration algorithm was verified through numerical experiments. For the LASSO problem, compared with the PIPL-ADMM algorithm and the PD-ADMM algorithm, the APIPL-ADMM algorithm has significantly fewer iterations, shorter running time, and better computing efficiency.

TABLE 4. The average number of iterations and runtime of the APIPL-ADMM and PD-ADMM

γ	$m \times n$	PIPL-ADMM Iter.1	PIPL-ADMM Time.1	APIPL-ADMM Iter.2	APIPL-ADMM Time.2	$\frac{Iter.2}{Iter.1}$	$\frac{Time2}{Time1}$
	900×3000	51	2.86	15	0.89	0.29	0.31
	1050×3500	54	3.90	15	1.07	0.28	0.27
$\gamma = 0.6$	1200×4000	45	4.40	15	1.47	0.33	0.33
	1350×4500	44	5.68	16	2.00	0.36	0.35
	1500×5000	43	6.89	17	2.58	0.40	0.37
	900×3000	52	2.80	14	0.80	0.27	0.29
	1050×3500	55	4.10	14	1.13	0.25	0.27
$\gamma = 0.65$	1200×4000	46	4.51	15	1.46	0.33	0.32
	1350×4500	45	5.97	14	1.79	0.31	0.30
	1500×5000	44	6.95	15	2.34	0.34	0.34
	900×3000	53	2.92	11	0.60	0.21	0.20
	1050×3500	57	4.25	13	0.99	0.23	0.23
$\gamma = 0.7$	1200×4000	47	5.16	14	1.43	0.30	0.28
	1350×4500	46	6.80	13	1.85	0.28	0.27
	1500×5000	46	7.50	15	2.44	0.33	0.32
	900×3000	55	2.98	12	0.67	0.22	0.23
	1050×3500	59	4.52	14	1.07	0.24	0.24
$\gamma = 0.75$	1200×4000	49	5.96	13	1.65	0.27	0.28
	1350×4500	48	7.16	12	1.72	0.25	0.24
	1500×5000	47	7.18	15	2.31	0.32	0.32

Acknowledgments

The authors are grateful to the anonymous referees and the associated editor for their valuable comments and suggestions, which help to improve the paper greatly. This paper was partially supported by the National Natural Science Foundation of China (Nos: 12071379, 12126412, and 11871383), the Natural Science Foundation of Chongqing (Nos: cstc2021jcyj-msxmX0925 and cstc2022ycjh-bgzxm0097), and the Youth Top Talent Program of Chongqing Talents (CQYC2021059311).

REFERENCES

[1] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finiteelement approximation, Comput. Math. Appl. 2 (1976), 17-40.

[2] D. Gabay, Applications of the method of multipliers to variational inequalities, In: Fortin, M., Glowinski, R.(eds) Augmented Lagrange Methods: Applications to thw Solution of Boundary-valued Problems, pp. 299-331, North Holland: Amsterdam, 1983.

[3] B. He, X. Yuan, On the $O(1/t)$ convergence rate of the Douglas-Rachford alternating direction method, SIAM J. Numer. Anal. 50 (2012), 700-709.

[4] D. Davis, W. Yin, Convergence rate analysis of several splitting schemes, Technical report, UCLACAM Report (2014).

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011), 1-122.

[6] R.H. Chan, J.F. Yang, X.M. Yuan, Alternating direction method for image inpainting in wavelet domains, SIAM J. Imag. Sci. 4 (2011), 807-826.
[7] D. Han, X.M. Yuan, A note on the alternating direction method of multipliers, J. Optim. Theory Appl. 155 (2012), 227-238.
[8] D. Han, X.M. Yuan, W. Zhang, X. Cai, An ADM-based splitting method for separable convex programming, Comput. Optim. Appl. 54 (2013), 343-369.
[9] K. Huai, M. Ni, Z. Yu, X. Zheng, F. Ma, A generalized inexact Uzawa method for stable principal component pursuit problem with nonnegative constraints, Numer. Algo. 77 (2018), 653-674.
[10] Z. Lin, R. Liu, H. Li, Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning, Mach. Learn. 99 (2015), 287-325.
[11] M.K. Ng, P. Weiss, X.M. Yuan, Solving constrainted total-variation image restoration and reconstruction problems via alternating direction methods, SIAM J. Sci. Comput. 32 (2010), 2710-2736.
[12] Y. Wang, J.F. Yang, W. Yin, Y. Zhang, A new alternation minimization algorithm for total variation image reconstruction, SIAM J. Image. Sci. 1 (2008), 248-272.
[13] B.S. He, L.Z. Liao, D. Han, H. Yang, A new inexact alternating direction method for monotone variational inequalutues, Math. Program. 92 (2002), 103-118.
[14] J.F. Yang, X.M. Yuan, Linearized augmented Lagrangain and alternating direction methods for nuclear norm minimization, Math. Comp. 82 (2013), 301-329.
[15] X.Q. Zhang, M. Burger, S. Osher, A unified primal-dual algorithm framework based on Bergman iteration, J. Sci. Comput. 46 (2010), 20-46.
[16] B.S. He, F. Ma, X.M. Yuan, Linearized alternating direction method of multipliers via positive-indefinite proximal regularization for convex programming, Optimization-online, http://www.optimization-online.org/DB-HTML/2016/07/5569.html (2016).
[17] J.W. Chen, Y.Y. Wang, H.J. He, Y.B. Lv, Convergence analysis of positive-indefinite proximal ADMM with a Glowinski's relaxation factor, Numer. Algo. 83 (2020), 1415-1440.
[18] H. Li, Z.C. Lin, Accelerated alternating direction method of multipliers: An optimal $O(1/K)$ nonergodic analysis, J. Sci. Comput. 79 (2019), 671-699.
[19] Y. Ouyang, Y. Chen, G. Lan, E. Pasiliao, An accelerated linearized alternating direction method of multipliers, SIAM J. Imag. Sci. 7 (2015), 1588-1623.
[20] Z.C. Lin, H. Li, C. Fang, Accelerated Optimization for Machine Learning: First-Order Algorithms, Springer, Singapore, 2020.
[21] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1996.