Journal of Mathematical Research with Applications
Mar., 2021, Vol. 41, No. 2, pp. 204–220
DOI:10.3770/j.issn:2095-2651.2021.02.008
Http://jmre.dlut.edu.cn

An Inertial Alternating Direction Method of Multipliers for Solving a Two-Block Separable Convex Minimization Problem

Yang YANG, Yuchao TANG*
Department of Mathematics, Nanchang University, Jiangxi 330031, P. R. China

Abstract The alternating direction method of multipliers (ADMM) is a widely used method for solving many convex minimization models arising in signal and image processing. In this paper, we propose an inertial ADMM for solving a two-block separable convex minimization problem with linear equality constraints. This algorithm is obtained by making use of the inertial Douglas-Rachford splitting algorithm to the corresponding dual of the primal problem. We study the convergence analysis of the proposed algorithm in infinite-dimensional Hilbert spaces. Furthermore, we apply the proposed algorithm on the robust principal component analysis problem and also compare it with other state-of-the-art algorithms. Numerical results demonstrate the advantage of the proposed algorithm.

Keywords alternating direction method of multipliers; inertial method; Douglas-Rachford splitting algorithm

MR(2020) Subject Classification 65K05; 65K15; 90C25

1. Introduction

In this paper, we consider the convex optimization problem of two-block separable objective functions with linear equality constraints:

$$\begin{aligned} \min_{u\in H_1, v\in H_2} \quad & F(u)+G(v) \\ \text{s.t.} \quad & Mu+Nv=b, \end{aligned} \tag{1.1}$$

where $b\in H$, $M: H_1\to H$ and $N: H_2\to H$ are bounded linear operators, $F: H_1\to(-\infty,+\infty]$ and $G: H_2\to(-\infty,+\infty]$ are proper, convex and lower semi-continuous functions (not necessarily smooth), H, H_1 and H_2 are real Hilbert spaces. Specifically, $H=R^p$, $H_1=R^n$, $H_2=R^m$. Then $M\in R^{p\times n}$ and $N\in R^{p\times m}$. We use the notations of Hilbert spaces for generality. Many problems in signal and image processing, medical image reconstruction, machine learning, and many other areas are special case of (1.1). When $N=-I$ and $b=0$, then (1.1) degenerates into

Received March 12, 2020; Accepted September 27, 2020
Supported by the National Natural Science Foundation of China (Grant Nos. 12061045; 12061046; 11661056; 11771198; 11771347; 91730306; 41390454; 11401293), the China Postdoctoral Science Foundation (Grant No. 2015M571989) and the Jiangxi Province Postdoctoral Science Foundation (Grant No. 2015KY51)
* Corresponding author
E-mail address: 1179916944@qq.com (Yang YANG); hhaaoo1331@163.com (Yuchao TANG)

an important special case of it as follows,

$$\begin{aligned} \min_{u\in H_1, v\in H_2} \quad & F(u)+G(v) \\ \text{s.t.} \quad & Mu = v, \end{aligned} \tag{1.2}$$

which is equivalent to

$$\min_{u\in H_1} F(u)+G(Mu). \tag{1.3}$$

The alternating direction method of multipliers (ADMM) is a popular way to solve (1.1) and (1.2). It has been attracted much attention because of its simplicity and efficiency. The ADMM can be dated back to the work of Glowinski and Marroco [1], and Gabay and Mercier [2]. The classical formulation of the ADMM for solving (1.1), which could be presented below:

$$\begin{cases} u^{k+1} = \arg\min\limits_{u}\{F(u)+\langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu+Nv^k-b\|^2\}, \\ v^{k+1} = \arg\min\limits_{v}\{G(v)+\langle y^k, Nv\rangle + \frac{\gamma}{2}\|Mu^{k+1}+Nv-b\|^2\}, \\ y^{k+1} = y^k + \gamma(Mu^{k+1}+Nv^{k+1}-b), \end{cases} \tag{1.4}$$

where $\gamma > 0$. It is well-known that the ADMM can be interpreted as an application of the Douglas-Rachford splitting algorithm to the dual of the problem (1.1). See, for instance [3]. It is worth mentioning that the famous split Bregman method [4] is also equivalent to the ADMM. See, e.g., [5]. A comprehensive review of the ADMM with applications in various convex optimization problems can be found in [6]. For the convergence and convergence rate analysis of ADMM for solving (1.1) and (1.3), we refer interested readers to [7–10] for more details. Also, many efforts have been tried to extend the ADMM for solving multi-block separable convex minimization problems, see for instance [11–13]. In this paper, we focus on the general case of the two-block separable convex minimization problem (1.1).

The generalized ADMM (GADMM) proposed by Eckstein and Bertsekas [3] is an efficient and simple acceleration scheme of the classical ADMM (1.4) for solving (1.3). It is easy to extend the GADMM to solve (1.1), and the iterative scheme of the GADMM reads as

$$\begin{cases} u^{k+1} = \arg\min\limits_{u}\{F(u)+\langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu+Nv^k-b\|^2\}, \\ v^{k+1} = \arg\min\limits_{v}\{G(v)+\langle y^k, Nv\rangle + \frac{\gamma}{2}\|N(v-v^k)+\lambda_k(Mu^{k+1}+Nv^k-b)\|^2\}, \\ y^{k+1} = y^k + \gamma(N(v^{k+1}-v^k)+\lambda_k(Mu^{k+1}+Nv^k-b)), \end{cases} \tag{1.5}$$

where $\gamma > 0$ and $\lambda_k \in (0,2)$. Let $\lambda_k = 1$. Then the GADMM (1.5) reduces to the classical ADMM (1.4). There are many works demonstrating that the GADMM (1.5) can numerically accelerate the classical ADMM (1.4) with λ_k belonging to $(1,2)$. See, for example [9, 14].

In recent years, the inertial method becomes more and more popular. It can be used to accelerate the first-order method for solving nonsmooth convex optimization problems. It is closely related to the famous Nesterov's accelerated method, which utilizes the current iteration information and the previous iteration information to update the new iteration. Many inertial algorithms have been proposed and studied, such as inertial proximal point algorithm [15, 16], inertial forward-backward splitting algorithm [17, 18], inertial forward-backward-forward splitting algorithm [19], inertial three-operator splitting algorithm [20], etc. There are also several attempts to introduce the inertial method to the ADMM. In particular, Chen et al. [21] proposed an inertial proximal ADMM by combining the proximal ADMM [22] and the inertial proximal point algorithm [15]. The detail of the inertial proximal ADMM is presented below.

$$\begin{cases} (\bar{u}^k, \bar{v}^k, \bar{y}^k) = (u^k, v^k, y^k) + \alpha_k(u^k - u^{k-1}, v^k - v^{k-1}, y^k - y^{k-1}), \\ u^{k+1} = \arg\min_u \{F(u) + \langle \bar{y}^k, Mu\rangle + \frac{\gamma}{2}\|Mu + N\bar{v}^k - b\|^2 + \frac{1}{2}\|u - \bar{u}^k\|_S^2\}, \\ y^{k+1} = \bar{y}^k + \gamma(Mu^{k+1} + N\bar{v}^k - b), \\ v^{k+1} = \arg\min_v \{G(v) + \langle y^{k+1}, Nv\rangle + \frac{\gamma}{2}\|Mu^{k+1} + Nv - b\|^2 + \frac{1}{2}\|v - \bar{v}^k\|_T^2\}, \end{cases} \tag{1.6}$$

where $\{\alpha_k\}$ are usually called inertial parameters, S and T are symmetric and positive semidefinite matrices, and $\gamma > 0$ is a penalty parameter. Let the matrices $S = T = 0$ and $\alpha_k = 0$, then the inertial proximal ADMM (1.6) recovers the following ADMM type algorithm, which is studied in [14].

$$\begin{cases} u^{k+1} = \arg\min_u \{F(u) + \langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu + Nv^k - b\|^2\}, \\ y^{k+1} = y^k + \gamma(Mu^{k+1} + Nv^k - b), \\ v^{k+1} = \arg\min_v \{G(v) + \langle y^{k+1}, Nv\rangle + \frac{\gamma}{2}\|Mu^{k+1} + Nv - b\|^2\}. \end{cases} \tag{1.7}$$

The difference between (1.4) and (1.7) is that the update order of the former is $u^{k+1} \to v^{k+1} \to y^{k+1}$, but the update order of the later is $u^{k+1} \to y^{k+1} \to v^{k+1}$. In contrast, Boţ and Csetnek [23] proposed an inertial ADMM for solving the convex optimization problem (1.3), which was based on the inertial Douglas-Rachford splitting algorithm [24]. It takes the form of

$$\begin{cases} u^{k+1} = \arg\min_u \{F(u) + \langle y^k - \alpha_k(y^k - y^{k-1}) - \gamma\alpha_k(v^k - v^{k-1}), Mu\rangle + \\ \qquad \frac{\gamma}{2}\|Mu - v^k\|^2\}, \\ \overline{v}^{k+1} = \alpha_{k+1}\lambda_k(Mu^{k+1} - v^k) + \frac{(1-\lambda_k)\alpha_k\alpha_{k+1}}{\gamma}(y^k - y^{k-1} + \gamma(v^k - v^{k-1})), \\ v^{k+1} = \arg\min_v \{G(v + \overline{v}^{k+1}) + \langle -y^k - (1-\lambda_k)\alpha_k(y^k - y^{k-1} + \gamma(v^k - v^{k-1})), v\rangle + \\ \qquad \frac{\gamma}{2}\|v - \lambda_k Mu^{k+1} - (1-\lambda_k)v^k\|^2\}, \\ y^{k+1} = y^k + \gamma(\lambda_k Mu^{k+1} + (1-\lambda_k)v^k - v^{k+1}) + \\ \qquad (1-\lambda_k)\alpha_k(y^k - y^{k-1} + \gamma(v^k - v^{k-1})). \end{cases} \tag{1.8}$$

Let $\alpha_k = 0$ and $\lambda_k = 1$, then (1.8) becomes the classical ADMM for solving the convex optimization problem (1.3). Boţ and Csetnek analyzed the convergence of the sequences generated by the inertial ADMM (1.8) under mild conditions on the parameters α_k and λ_k.

The inertial proximal ADMM (1.6) and the inertial ADMM (1.8) are two different algorithms. Since the inertial proximal ADMM (1.6) is derived from the inertial proximal point algorithm with a special precondition matrix, which is applied to the KKT condition of the convex minimization problem (1.1). While the inertial ADMM (1.8) is obtained by applying the inertial Douglas-Rachford splitting algorithm to the dual of the problem (1.3).

The purpose of this paper is to introduce an inertial ADMM for solving the general two-block separable convex optimization problem (1.1). We prove the convergence of the sequences generated by the proposed inertial ADMM. It is worth mentioning that the convex minimization problem (1.1) can also be rewritten as (1.3), then we can apply the inertial ADMM (1.8) to solve it. Compared with the inertial ADMM (1.8), the iteration scheme of the proposed algorithm is more simple. In particular, we present the relationship between the proposed algorithm

(Algorithm 1) and the inertial ADMM (1.8) (See Remark 3.3). Finally, we conduct numerical experiments on robust principal component analysis (RPCA) problem and compare the proposed algorithm with the classical ADMM (1.4), the GADMM (1.5), and the inertial proximal ADMM (1.6) to demonstrate the advantage of the proposed algorithm.

We would like to highlight the contributions of this paper. (i) An inertial ADMM is developed to solve the convex minimization problem (1.1); (ii) The convergence of the proposed inertial ADMM is studied in infinite-dimensional Hilbert space; (iii) The effectiveness and efficiency of the proposed inertial ADMM are demonstrated by applying to the RPCA problem.

The structure of this paper is as follows. In the next section, we summarize some notations and definitions that will be used in the following sections. We also recall the inertial Douglas-Rachford splitting algorithm. In Section 3, we introduce an inertial ADMM and study its convergence results in detail. In Section 4, some numerical experiments for solving the RPCA problem (4.2) are conducted to demonstrate the efficiency of the proposed algorithm. Finally, we give some conclusions.

2. Preliminaries

In this section, we recall some concepts of monotone operator theory and convex analysis in Hilbert spaces. Most of them can be found in [25]. Let H_1 and H_2 be real Hilbert spaces with inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$. $x_k \rightharpoonup x$ stands for $\{x_k\}$ converging weakly to x, and $x_k \to x$ stands for $\{x_k\}$ converging strongly to x. Let $A : H_1 \to H_2$ be a linear continuous operator and its adjoint operator be $A^* : H_2 \to H_1$ is the unique operator that satisfies $\langle A^* y, x \rangle = \langle y, Ax \rangle$ for all $x \in H_1$ and $y \in H_2$.

Let $A : H \to 2^H$ be a set-valued operator. We denote by $\operatorname{zer} A = \{x \in H : 0 \in Ax\}$ its set of zeros, by $\operatorname{Gra} A = \{(x, u) \in H \times H : u \in Ax\}$ its graph and by $A^{-1} : H \to 2^H$ its inverse operator, where $(u, x) \in \operatorname{Gra} A^{-1}$ if and only if $(x, u) \in \operatorname{Gra} A$. We say that A is monotone if $\langle x - y, u - v \rangle \geq 0$, for all $(x, u), (y, v) \in \operatorname{Gra} A$. A is said to be maximally monotone if its graph is not contained in the graph of any other monotone operator. Letting $\gamma > 0$, the resolvent of γA is defined by

$$J_{\gamma A} = (I + \gamma A)^{-1}. \tag{2.1}$$

Moreover, if A is maximally monotone, then $J_{\gamma A} : H \to H$ is single-valued and maximally monotone.

The operator A is uniformly monotone with modulus $\phi_A : R_+ \to [0, +\infty)$, if ϕ_A is increasing, ϕ_A vanishes only at 0, and

$$\langle x - y, u - v \rangle \geq \phi_A(\|x - y\|), \tag{2.2}$$

for all $(x, u), (y, v) \in \operatorname{Gra} A$. Moreover, if $\phi_A = \gamma(\cdot)^2$ $(\gamma > 0)$, then A is γ-strongly monotone.

For a function $f : H \to \bar{R}$, where $\bar{R} := R \bigcup \{+\infty\}$, We define $\operatorname{dom} f = \{x \in H : f(x) < +\infty\}$ as its effective domain and say that f is proper if $\operatorname{dom} f \neq \emptyset$ and $f(x) \neq -\infty$ for all $x \in H$. Let $f : H \to \bar{R}$. The conjugate of f is $f^* : H \to \bar{R} : u \mapsto \sup_{x \in H} \{\langle x, u \rangle - f(x)\}$. We denote by $\Gamma_0(H)$ the family of proper, convex and lower semi-continuous extended real-valued functions defined on H. Let $f \in \Gamma_0(H)$. The subdifferential of f is the set-valued operator $\partial f(x) := \{v \in H : f(y) \geq f(x) + \langle v, y - x\rangle \, \forall y \in H\}$. Moreover, ∂f is a maximally monotone operator and it holds $(\partial f)^{-1} = \partial f^*$. For every $x \in H$ and arbitrary $\gamma > 0$, we have

$$\mathrm{Prox}_{\gamma f}(x) = \arg\min_y\{\gamma f(y) + \frac{1}{2}\|y - x\|^2\}. \tag{2.3}$$

Here $\mathrm{Prox}_{\gamma f}(x)$ is the proximal point of parameter γ of f at x. And the proximal point operator of f satisfies $\mathrm{Prox}_{\gamma f}(x) = (I + \gamma\partial f)^{-1}x = J_{\gamma\partial f}x$.

Let $f \in \Gamma_0(H)$. Then f is uniformly convex with modulus $\phi : R_+ \to [0, +\infty)$ if ϕ is increasing, ϕ vanishes only at 0, and

$$f(\alpha x + (1-\alpha)y) + \alpha(1-\alpha)\phi(\|x - y\|) \leq \alpha f(x) + (1-\alpha)f(y), \tag{2.4}$$

for all $x, y \in \mathrm{dom}\, f$, $\forall\alpha \in (0, 1)$. If (2.4) holds with $\phi = \frac{\beta}{2}(\cdot)^2$ for some $\beta \in (0, +\infty)$, then f is β-strongly convex. This also means that ∂f is β-strongly monotone (if f is uniformly convex, then ∂f is uniformly monotone).

At the end of this section, we recall the main results of the inertial Douglas-Rachford splitting algorithm in [24].

Theorem 2.1 ([24]) *Let $A, B : H \to 2^H$ be maximally monotone operators. Assume $\mathrm{zer}(A + B) \neq \emptyset$. For any given $w^0, w^1 \in H$, define the iterative sequences as follows:*

$$\begin{cases} y^k = J_{\gamma B}(w^k + \alpha_k(w^k - w^{k-1})), \\ x^k = J_{\gamma A}(2y^k - w^k - \alpha_k(w^k - w^{k-1}), \\ w^{k+1} = w^k + \alpha_k(w^k - w^{k-1}) + \lambda_k(x^k - y^k), \end{cases} \tag{2.5}$$

where $\gamma > 0$, $\{\alpha_k\}_{k\geq1}$ is nondecreasing with $\alpha_1 = 0$ and $0 \leq \alpha_k \leq \alpha < 1$ for every $k \geq 1$ and $\lambda, \delta, \sigma > 0$ are such that

$$\delta > \frac{\alpha^2(1+\alpha) + \alpha\sigma}{1 - \alpha^2}, \quad \text{and} \quad 0 < \lambda \leq \lambda_k \leq 2\frac{\delta - \alpha[\alpha(1+\alpha) + \alpha\delta + \sigma]}{\delta[1 + \alpha(1+\alpha) + \alpha\delta + \sigma]}, \quad \forall k \geq 1. \tag{2.6}$$

Then there exists $x \in H$ such that the following statements are true:

(1) $J_{\gamma B}x \in \mathrm{zer}(A + B)$;

(2) $\Sigma_{k\in N}\|w^{k+1} - w^k\|^2 < +\infty$;

(3) $\{w^k\}_{k\geq0}$ converges weakly to x;

(4) $y^k - x^k \to 0$ as $k \to +\infty$;

(5) $\{y^k\}_{k\geq1}$ converges weakly to $J_{\gamma B}x$;

(6) $\{x^k\}_{k\geq1}$ converges weakly to $J_{\gamma B}x = J_{\gamma A}(2J_{\gamma B}x - x)$;

(7) If A or B is uniformly monotone, then $\{y^k\}_{k\geq1}$ and $\{x^k\}_{k\geq1}$ converges strongly to unique point in $\mathrm{zer}(A + B)$.

Remark 2.2 According to [24], the condition $\alpha_1 = 0$ in the above theorem can be replaced by the assumption $w^0 = w^1$.

3. Inertial ADMM for solving the two-block separable convex minimization problem

In this section, we present the main results of this paper. Let H, H_1 and H_2 be real Hilbert spaces, $F \in \Gamma_0(H_1)$, $G \in \Gamma_0(H_2)$, $b \in H$, $M : H_1 \to H$ and $N : H_2 \to H$ are linear continuous operators. The dual problem of (1.1) is

$$\max_{y \in H} -F^*(-M^*y) - G^*(-N^*y) - \langle y, b\rangle, \tag{3.1}$$

where F^* and G^* are the Fenchel-conjugate functions of F and G, respectively. We consider solving the convex optimization problems (1.1) and its dual problem (3.1). Let $v(P)$ and $v(D)$ be the optimal objective values of the above two problems respectively, the situation $v(P) \geq v(D)$, called in the literature weak duality, always holds. We introduce the Attouch-Brézies condition, that is

$$0 \in \operatorname{sqri}(M(\operatorname{dom} F) - N(\operatorname{dom} G)). \tag{3.2}$$

For arbitrary convex set $C \subseteq H$, we define its strong quasi-relative interior as

$$\operatorname{sqri} C := \{x \in C : \cup_{\lambda>0}\lambda(C - x) \text{ is a closed linear subspace of } H\}. \tag{3.3}$$

If (3.2) holds, then we have strong duality, which means that $v(P) = v(D)$ and (3.1) has an optimal solution.

Next, we introduce the main algorithm in this paper.

Algorithm 1 An inertial alternating direction method of multipliers (iADMM)

Input: For arbitrary $y^1 \in H$, $v^1 \in H_2$, $p^1 = 0$, choose γ, α_k and λ_k.

For $k = 1, 2, 3, \ldots$, compute

1: $u^{k+1} = \arg\min\limits_{u}\{F(u) + \langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu + Nv^k - b\|^2\}$;

2: $v^{k+1} = \arg\min\limits_{v}\{G(v) + \langle y^k + \alpha_{k+1}p^k, Nv\rangle + \frac{\gamma}{2}\|N(v - v^k) + (1+\alpha_{k+1})\lambda_k(Mu^{k+1} + Nv^k - b)\|^2\}$;

3: $y^{k+1} = y^k + \alpha_{k+1}p^k + \gamma[N(v^{k+1} - v^k) + (1 + \alpha_{k+1})\lambda_k(Mu^{k+1} + Nv^k - b)]$;

4: $p^{k+1} = \frac{\alpha_{k+1}}{1+\alpha_{k+1}}[p^k + (y^{k+1} - y^k) - \gamma N(v^{k+1} - v^k)]$;

Stop when a given stopping criterion is met.

Output: u^{k+1}, v^{k+1} and y^{k+1}.

In order to analyze the convergence of Algorithm 1, we define the Lagrangian function of problem (1.1) as follows:

$$L(u, v, y) = F(u) + G(v) + \langle y, Mu + Nv - b\rangle, \tag{3.4}$$

where y is a Lagrange multiplier. Assuming (u^*, v^*) is an optimal solution of the optimization problem (1.1), there exits a vector y^*, according to KKT condition, we have

$$\begin{aligned} &0 \in \partial F(u^*) + M^*y^*,\\ &0 \in \partial G(v^*) + N^*y^*,\\ &Mu^* + Nv^* - b = 0. \end{aligned} \tag{3.5}$$

Moreover, point pairs (u^*, v^*, y^*) are saddle points of Lagrange function, that is

$$L(u^*, v^*, y) \leq L(u^*, v^*, y^*) \leq L(u, v, y^*), \quad \forall (u, v, y) \in H_1 \times H_2 \times H. \tag{3.6}$$

In order to analyze the convergence of the proposed Algorithm 1 in Hilbert spaces, we show that the iterative sequences generated by Algorithm 1 are instances of the inertial Douglas-Rachford splitting algorithm (2.5) applied to the dual problem (3.1). In detail, we show that Algorithm 1 could be derived from the inertial Douglas-Rachford splitting algorithm (2.5). Then we use Theorem 2.1 to obtain the convergence of the proposed Algorithm 1. Now, we are ready to present the main convergence theorem of Algorithm 1.

Theorem 3.1 *Assuming (1.1) has an optimal solution and the condition (3.2) is satisfied. Let the bounded linear operators M and N satisfy the condition that $\exists\theta_1 > 0, \theta_2 > 0$ such that $\|Mx\| \geq \theta_1\|x\|$ and $\|Nx\| \geq \theta_2\|x\|$, for all $x \in H$. Consider the sequence generated by Algorithm 1. Let $\gamma > 0$, $\{\alpha_k\}_{k\geq 1}$ nondecreasing with $0 \leq \alpha_k \leq \alpha < 1$, $\{\lambda_k\}_{k\geq 1}$ and $\lambda, \sigma, \delta > 0$ such that*

$$\delta > \frac{\alpha^2(1+\alpha)+\alpha\sigma}{1-\alpha^2} \text{ and } 0 < \lambda \leq \lambda_k \leq 2\frac{\delta - \alpha[\alpha(1+\alpha)+\alpha\delta+\sigma]}{\delta[1+\alpha(1+\alpha)+\alpha\delta+\sigma]}, \quad \forall k \geq 1.$$

Then there exists a point pair (u^*, v^*, y^*), which is the saddle point of Lagrange function, where (u^*, v^*) is the optimal solution of (1.1), y^* is the optimal solution of (3.1), and $v(P) = v(D)$. The following statements are true:

(i) $(u^k, v^k)_{k\geq 1}$ converges weakly to (u^*, v^*);

(ii) $(Mu^{k+1} + Nv^k)_{k\geq 1}$ converges strongly to b;

(iii) $(y^k)_{k\geq 1}$ converges weakly to y^*;

(iv) If F^* or G^* is uniformly convex, then $(y^k)_{k\geq 1}$ converges strongly to the unique optimal solution of (D);

(v) $\lim_{k\to+\infty} F(u^{k+1}) + G(v^k) = v(P) = v(D) = \lim_{k\to+\infty}(-F^*(-M^*x^k) - G^*(-N^*y^k) - \langle y^k, b\rangle)$, where the sequence $(x^k)_{k\geq 1}$ is defined by

$$x^k = y^k + \gamma M u^{k+1} + \gamma N v^k - \gamma b,$$

and $(x^k)_{k\geq 1}$ converges weakly to y^*.

Proof Let

$$A = \partial(F^* \circ (-M^*)), \quad \text{and} \quad B = \partial(G^* \circ (-N^*)) + b. \tag{3.7}$$

From the first step of iteration scheme (2.5), we have

$$\begin{aligned} y^k &= J_{\gamma B}(w^k + \alpha_k(w^k - w^{k-1})) \\ &= \mathrm{Prox}_{\gamma(G^*(-N^*)+\langle\cdot,b\rangle)}(w^k + \alpha_k(w^k - w^{k-1})) \\ &= \arg\min_y\{\gamma(G^*(-N^*y) + \langle y, b\rangle) + \frac{1}{2}\|y - w^k - \alpha_k(w^k - w^{k-1})\|^2\}. \end{aligned} \tag{3.8}$$

By the first-order optimality condition, we obtain from (3.8) that

$$0 \in -\gamma N\partial G^*(-N^*y^k) + \gamma b + y^k - w^k - \alpha_k(w^k - w^{k-1}). \tag{3.9}$$

We introduce the sequence $\{v^k\}_{k\geq 1}$ by

$$v^k \in \partial G^*(-N^*y^k), \tag{3.10}$$

then, we have

$$0 = -\gamma N v^k + \gamma b + y^k - w^k - \alpha_k(w^k - w^{k-1}) \text{ and } -N^* y^k \in \partial G(v^k). \tag{3.11}$$

From the second step of iteration scheme (2.5), we have

$$\begin{aligned} x^k &= J_{\gamma A}(2y^k - w^k - \alpha_k(w^k - w^{k-1})) \\ &= \mathrm{Prox}_{\gamma F^*(-M^*)}(2y^k - w^k - \alpha_k(w^k - w^{k-1})) \\ &= \arg\min_x \{\gamma F^*(-M^* x) + \frac{1}{2}\|x - 2y^k + w^k + \alpha_k(w^k - w^{k-1})\|^2\}. \end{aligned} \tag{3.12}$$

According to the first-order optimality condition, there are also

$$0 \in -\gamma M \partial F^*(-M^* x^k) + x^k - 2y^k + w^k + \alpha_k(w^k - w^{k-1}). \tag{3.13}$$

Again we introduce a new sequence $\{u^{k+1}\}_{k\geq 1}$ by

$$u^{k+1} \in \partial F^*(-M^* x^k), \tag{3.14}$$

we can get

$$0 = -\gamma M u^{k+1} + x^k - 2y^k + w^k + \alpha_k(w^k - w^{k-1}) \text{ and } -M^* x^k \in \partial F(u^{k+1}). \tag{3.15}$$

From the first formula of (3.11) and (3.15), we obtain

$$x^k - y^k = \gamma M u^{k+1} + \gamma N v^k - \gamma b. \tag{3.16}$$

Combining the second formula of (3.15) and (3.16), we have

$$\begin{aligned} &0 \in \partial F(u^{k+1}) + M^* x^k \\ &\Leftrightarrow 0 \in \partial F(u^{k+1}) + M^*(y^k + \gamma M u^{k+1} + \gamma N v^k - \gamma b). \end{aligned}$$

Therefore, it is clear that

$$u^{k+1} = \arg\min_u \{F(u) + \langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu + Nv^k - b\|^2\}. \tag{3.17}$$

This is the first step of Algorithm 1.

Let $p^k = \alpha_k(w^k - w^{k-1})$, then the first formula of (3.11) can be rewritten as

$$w^k = y^k + \gamma b - p^k - \gamma N v^k. \tag{3.18}$$

Furthermore, we have

$$\begin{aligned} p^{k+1} &= \alpha_{k+1}(w^{k+1} - w^k) \\ &= \alpha_{k+1}(y^{k+1} + \gamma b - p^{k+1} - \gamma N v^{k+1} - y^k - \gamma b + p^k + \gamma N v^k), \end{aligned}$$

hence

$$p^{k+1} = \alpha_{k+1}(w^{k+1} - w^k) = \frac{\alpha_{k+1}}{1 + \alpha_{k+1}}(p^k + (y^{k+1} - y^k) - \gamma N(v^{k+1} - v^k)). \tag{3.19}$$

This is the fourth step of Algorithm 1.

From the third step of iteration scheme (2.5), the first formula of (3.11), (3.16) and (3.19), we get

$$y^{k+1} = y^k + \alpha_{k+1} p^k + \gamma[N(v^{k+1} - v^k) + (1 + \alpha_{k+1})\lambda_k(Mu^{k+1} + Nv^k - b)]. \tag{3.20}$$

This is the third step of Algorithm 1.

Combining the second formula of (3.15) and (3.20), we obtain

$$0 \in \partial G(v^{k+1}) + N^* y^{k+1};$$
$$0 \in \partial G(v^{k+1}) + N^*(y^k + \alpha_{k+1} p^k + \gamma(N(v^{k+1} - v^k) + (1 + \alpha_{k+1})\lambda_k(Mu^{k+1} + Nv^k - b))).$$

Therefore, it is clear that

$$\begin{aligned} v^{k+1} =& \arg\min_v \{G(v) + \langle y^k + \alpha_{k+1} p^k, Nv\rangle + \frac{\gamma}{2}\|N(v - v^k)+ \\ & (1 + \alpha_{k+1})\lambda_k(Mu^{k+1} + Nv^k - b)\|^2\}. \end{aligned} \tag{3.21}$$

This is the second step of Algorithm 1.

Therefore, Algorithm 1 is equivalent to the inertial Douglas-Rachford splitting algorithm for the dual problem of the original problem (1.1).

Next, we prove the convergence of Theorem 3.1. According to Theorem 2.1, there exists $\bar{w} \in H$ such that

$$w^k \rightharpoonup \bar{w} \text{ as } k \to +\infty, \tag{3.22}$$

$$w^{k+1} - w^k \to 0 \text{ as } k \to +\infty, \tag{3.23}$$

$$y^k \rightharpoonup J_{\gamma B}\bar{w} \text{ as } k \to +\infty, \tag{3.24}$$

$$x^k \rightharpoonup J_{\gamma B}\bar{w} = J_{\gamma A}(2J_{\gamma B}\bar{w} - \bar{w}) \text{ as } k \to +\infty, \tag{3.25}$$

$$y^k - x^k \to 0 \text{ as } k \to +\infty. \tag{3.26}$$

(i) From the first formula of (3.15), we have

$$\gamma M u^{k+1} = x^k - 2y^k + w^k + \alpha_k(w^k - w^{k-1}),$$

and from (3.22), (3.23), (3.24) and (3.26), we can get

$$Mu^{k+1} \rightharpoonup \frac{1}{\gamma}(\bar{w} - J_{\gamma B}\bar{w}) \text{ as } k \to +\infty. \tag{3.27}$$

Moreover, by the first formula of (3.11), we have

$$\gamma N v^k = \gamma b + y^k - w^k - \alpha_k(w^k - w^{k-1}),$$

and then through (3.22), (3.23) and (3.24), we derive that

$$Nv^k \rightharpoonup \frac{1}{\gamma}(J_{\gamma B}\bar{w} - \bar{w}) + b \text{ as } k \to +\infty. \tag{3.28}$$

Let

$$Mu^* = \frac{1}{\gamma}(\bar{w} - J_{\gamma B}\bar{w}) \text{ and } Nv^* = \frac{1}{\gamma}(J_{\gamma B}\bar{w} - \bar{w}) + b. \tag{3.29}$$

Since the operators M and N satisfy the conditions $\|Mx\| \geq \theta_1\|x\|$ and $\|Nx\| \geq \theta_2\|x\|$ for all $x \in H$, according to (3.27) and (3.28), there exist u^*, v^* such that

$$u^k \rightharpoonup u^* \text{ and } v^k \rightharpoonup v^* \text{ as } k \to +\infty, \tag{3.30}$$

and

$$Mu^* + Nv^* = b. \tag{3.31}$$

(ii) It follows from (3.16) and (3.26) that

$$Mu^{k+1} + Nv^k \to b \text{ as } k \to +\infty. \tag{3.32}$$

(iii) Let

$$y^* = J_{\gamma B}\bar{w}. \tag{3.33}$$

From (3.24), we have

$$y^k \rightharpoonup y^*. \tag{3.34}$$

(iv) According to (3.33), we have

$$\bar{w} \in y^* + \gamma B y^*, \tag{3.35}$$

thus

$$Nv^* = \frac{1}{\gamma}(J_{\gamma B}\bar{w} - \bar{w}) + b = \frac{1}{\gamma}(y^* - \bar{w}) + b \in -By^* + b. \tag{3.36}$$

Substituting $B = \partial(G^* \circ (-N^*)) + b$ to (3.36), we have

$$Nv^* \in N\partial G^*(-N^*y^*) \Leftrightarrow 0 \in \partial G(v^*) + N^*y^*. \tag{3.37}$$

Further, by (3.25) and (3.33), we obtain

$$y^* - \bar{w} \in \gamma A y^*, \tag{3.38}$$

that is

$$Mu^* = \frac{1}{\gamma}(\bar{w} - J_{\gamma B}\bar{w}) = \frac{1}{\gamma}(\bar{w} - y^*) \in -Ay^*. \tag{3.39}$$

Substituting $A = \partial(F^* \circ (-M^*))$ to (3.39), we have

$$Mu^* \in M\partial F^*(-M^*y^*) \Leftrightarrow 0 \in \partial F(u^*) + M^*y^*. \tag{3.40}$$

According to (3.31), (3.37) and (3.40), we prove that the point pair (u^*, v^*, y^*) satisfies the optimality condition (3.5). Theorem 3.1 (iv) can be obtained directly from Theorem 2.1 (7).

(v) We know that F and G are weakly lower semi-continuous (since F and G are convex) and therefore, from (i), we have

$$\begin{aligned}\liminf_{k\to+\infty}(F(u^{k+1}) + G(v^k)) &\geq \liminf_{k\to+\infty} F(u^{k+1}) + \liminf_{k\to+\infty} G(v^k) \\ &\geq F(u^*) + G(v^*) = v(P).\end{aligned} \tag{3.41}$$

We deduce from the second formula of (3.11) and (3.15) that

$$G(v^*) \geq G(v^k) + \langle v^* - v^k, -N^*y^k\rangle; \tag{3.42}$$

$$F(u^*) \geq F(u^{k+1}) + \langle u^* - u^{k+1}, -M^*x^k\rangle. \tag{3.43}$$

Summing up (3.42) and (3.43), we obtain

$$v(P) \geq F(u^{k+1}) + G(v^k) + \langle u^* - u^{k+1}, -M^*x^k\rangle + \langle v^* - v^k, -N^*y^k\rangle, \tag{3.44}$$

that is

$$\begin{aligned} v(P) \geq & F(u^{k+1}) + G(v^k) + \langle Mu^* - Mu^{k+1}, y^k - x^k \rangle + \\ & \langle Mu^* + Nv^* - Mu^{k+1} - Nv^k, -y^k \rangle. \end{aligned} \tag{3.45}$$

From (i), (ii), (iii), (3.26) and (3.31), we have

$$\limsup_{k\to+\infty}(F(u^{k+1}) + G(v^k)) \leq v(P). \tag{3.46}$$

Combining (3.41) and (3.46), we prove the first part of Theorem 3.1 (v).

Again from the second formula of (3.11) and (3.15), we get

$$G(v^k) + G^*(-N^*y^k) = \langle -N^*y^k, v^k \rangle; \tag{3.47}$$

$$F(u^{k+1}) + F^*(-M^*x^k) = \langle -M^*x^k, u^{k+1} \rangle. \tag{3.48}$$

Summing up (3.47) and (3.48), we have

$$\begin{aligned} F(u^{k+1}) + G(v^k) &= -F^*(-M^*x^k) - G^*(-N^*y^k) + \langle -x^k, Mu^{k+1} \rangle + \langle -y^k, Nv^k \rangle \\ &= -F^*(-M^*x^k) - G^*(-N^*y^k) + \langle y^k - x^k, Mu^{k+1} \rangle + \langle -y^k, Mu^{k+1} + Nv^k \rangle. \end{aligned}$$

Finally, taking into account (i), (ii), (3.26) and the first part of Theorem 3.1 (v), we obtain

$$\lim_{k\to+\infty}(-F^*(-M^*x^k) - G^*(-N^*y^k) - \langle y^k, b \rangle) = v(P) = v(D). \tag{3.49}$$

This completes the proof. □

Remark 3.2 Notice that, in finite-dimensional case, the assumption on M and N in Theorem 3.1 means that M and N are matrices with full column rank.

Remark 3.3 As we can see, when $\alpha_k \equiv 0$, the iterative sequences of Algorithm 1 reduces to the GADMM (1.5).

Suppose that the matrix N is full column rank, let $b - Nv = z$, then $Mu = z$ and $v = (N^*N)^{-1}N^*(b-z)$. Let $H(z) = G((N^*N)^{-1}N^*(b-z)) = G(v)$, then problem (1.1) is equivalent to problem (1.3), i.e., $\min\limits_{u\in H_1} F(u) + H(Mu)$. Therefore, we can directly apply the algorithm (1.8) and obtain the following iteration scheme:

$$\begin{cases} u^{k+1} = \arg\min\limits_{u}\{F(u) + \langle y^k - \alpha_k(y^k - y^{k-1} - \gamma N(v^k - v^{k-1})), Mu \rangle + \\ \qquad\qquad \frac{\gamma}{2}\|Mu + Nv^k - b\|^2\}, \\ \bar{v}^{k+1} = -\frac{\alpha_{k+1}}{\gamma}(N^*N)^{-1}N^*[\gamma\lambda_k(Mu^{k+1} + Nv^k - b) + \\ \qquad\qquad (1-\lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(v^k - v^{k-1}))], \\ v^{k+1} = \arg\min\limits_{v}\{G(v + \bar{v}^{k+1}) + \langle y^k + (1-\lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(v^k - v^{k-1})), Nv \rangle + \\ \qquad\qquad \frac{\gamma}{2}\|N(v - v^k) + \lambda_k(Mu^{k+1} + Nv^k - b)\|^2\}, \\ y^{k+1} = y^k + \gamma[N(v^{k+1} - v^k) + \lambda_k(Mu^{k+1} + Nv^k - b)] + \\ \qquad\qquad (1-\lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(v^k - v^{k-1})). \end{cases} \tag{3.50}$$

In the following, we prove that Algorithm 1 is equivalent to (3.50). In fact, according to (3.7) and (3.11), we have $\gamma Nv^k = \gamma b + y^k - w^k - \alpha_k(w^k - w^{k-1})$ and $\gamma Nv^k = y^k + \gamma b - w^k - p^k$, respectively. Let $\gamma Nt^k = \gamma b + y^k - w^k$ and $\gamma N\bar{t}^k = -\alpha_k(w^k - w^{k-1})$, we obtain

$$\gamma Nv^k = \gamma Nt^k + \gamma N\bar{t}^k,\ \ p^k = -\gamma N\bar{t}^k \quad \text{and} \quad \gamma N\bar{t}^k = -\alpha_k(y^k - y^{k-1} - \gamma N(t^k - t^{k-1})). \tag{3.51}$$

Substituting the first formula of (3.51) into step 1 of Algorithm 1, we obtain

$$u^{k+1} = \arg\min_u \{F(u) + \langle y^k, Mu\rangle + \frac{\gamma}{2}\|Mu + N(t^k + \bar{t}^k) - b\|^2\}, \tag{3.52}$$

and then substitute the last formula of (3.51) into (3.52), we get

$$u^{k+1} = \arg\min_u \{F(u) + \langle y^k - \alpha_k(y^k - y^{k-1} - \gamma N(t^k - t^{k-1})), Mu\rangle + \frac{\gamma}{2}\|Mu + Nt^k - b\|^2\}. \tag{3.53}$$

Substituting the (3.51) into step 4 of Algorithm 1, it is easy to get

$$\begin{aligned}
\bar{t}^{k+1} &= -\frac{1}{\gamma}(N^*N)^{-1}N^*p^{k+1} \\
&= -\frac{\alpha_{k+1}}{\gamma}(N^*N)^{-1}N^*(-\gamma N\bar{t}^k + \gamma\lambda_k(Mu^{k+1} + N(t^k + \bar{t}^k) - b)) \\
&= -\frac{\alpha_{k+1}}{\gamma}(N^*N)^{-1}N^*[\gamma\lambda_k(Mu^{k+1} + Nt^k - b) + \\
&\quad (1 - \lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(t^k - t^{k-1}))].
\end{aligned} \tag{3.54}$$

Similarly, substituting the (3.51) into step 2 of Algorithm 1, we have

$$\begin{aligned}
t^{k+1} =& \arg\min_t \{G(t + \bar{t}^{k+1}) + \langle y^k - \gamma\alpha_{k+1}N\bar{t}^k, Nt\rangle + \\
&\frac{\gamma}{2}\|N(t + \bar{t}^{k+1} - t^k - \bar{t}^k) + (1 + \alpha_{k+1})\lambda_k(Mu^{k+1} + N(t^k + \bar{t}^k) - b)\|^2\},
\end{aligned} \tag{3.55}$$

which implies that

$$\begin{aligned}
t^{k+1} =& \arg\min_t \{G(t + \bar{t}^{k+1}) + \langle y^k + (1 - \lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(t^k - t^{k-1})), Nt\rangle + \\
&\frac{\gamma}{2}\|N(t - t^k) + \lambda_k(Mu^{k+1} + Nt^k - b)\|^2\}.
\end{aligned} \tag{3.56}$$

Finally, combining steps 3 of Algorithm 1 and (3.51), we obtain

$$\begin{aligned}
y^{k+1} =& y^k + \gamma[N(t^{k+1} - t^k) + \lambda_k(Mu^{k+1} + Nt^k - b)] + \\
&(1 - \lambda_k)\alpha_k(y^k - y^{k-1} - \gamma N(t^k - t^{k-1})).
\end{aligned} \tag{3.57}$$

Compared with (3.50), it is obvious that the iteration scheme of Algorithm 1 is more concise.

4. Numerical experiments

In this section, we carry out simulation experiments and compare the proposed algorithm (Algorithm 1) with other state-of-the-art algorithms including the classical ADMM [26], GADMM [3], the inertial ADMM of Chen et al. [21] (IADMM_CCMY). All the experiments are conducted on 64-bit Windows 10 operating system with an Inter Core(TM) i5-6500 CPU and 8GB memory. All the codes are tested in MATLAB R2016a.

4.1. Robust principal component analysis (RPCA) problem

The robust principal component analysis (RPCA) problem was first introduced by Candès et al. [27], which can be formulated as the following optimization model:

$$\begin{aligned} \min_{u,v} \quad & \operatorname{rank}(u) + \mu\|v\|_0 \\ \text{s.t.} \quad & u + v = b, \end{aligned} \tag{4.1}$$

where $b \in R^{m\times m}$ is a given matrix, $\operatorname{rank}(u)$ denotes the rank of matrix u, $\|v\|_0$ is the number of the nonzeros of matrix v, and $\mu > 0$ is the penalty parameter balancing the low rank and sparsity. The objective function in (4.1) includes the rank of the matrix u and the ℓ_0-norm of matrix v, whose value is the number of nonzero elements in matrix v. The RPCA (4.1) seeks to decompose a matrix b into two parts: one is low-rank and the other is sparse. It has wide applications in image and video processing and many other fields. We refer interested readers to [28, 29] for a comprehensive review of RPCA and its variants. It is known that the original RPCA (4.1) is NP-hard. By using the convex relaxation technique, the rank function of the matrix is usually replaced by the nuclear norm of the matrix, and the ℓ_0-norm of the matrix is replaced by the ℓ_1-norm of the matrix. Therefore, we can get the following convex optimization model:

$$\begin{aligned} \min_{u,v} \quad & \|u\|_* + \mu\|v\|_1 \\ \text{s.t.} \quad & u + v = b, \end{aligned} \tag{4.2}$$

where $\|u\|_* = \sum_{k=1}^{r} \sigma_k(u)$ is the nuclear norm of the matrix and $\sigma_k(u)$ represents the k singular value of the matrix, and $\|v\|_1 = \sum_{ij} |v_{ij}|$. Under certain conditions, problem (4.2) is equivalent to (4.1). See for example [27,30]. In fact, let $F(u) = \|u\|_*$, $G(v) = \mu\|v\|_1$ and $M = N = I$. Then the RPCA (4.2) is a special case of the general problem (1.1). Therefore, the classical ADMM algorithm, GADMM algorithm and inertial ADMM algorithms (includes IADMM_CCMY and Algorithm 1) can be used to solve the convex optimization problem (4.2).

We follow [27] to generate the simulation data. In the experiment, a low rank matrix is randomly generated by the following method. Firstly, two long strip matrices $L = \operatorname{randn}(m, r)$ and $R = \operatorname{randn}(m, r)$ are randomly generated, and then $u^* = LR^T$ is calculated, where m and r are the order and rank of matrix u^*, respectively. At the same time, a sparse matrix v^* with uniform distribution of non-zero elements and uniform distribution of values between $[-500, 500]$ is generated. Finally, the target matrix is generated by $b = u^* + v^*$.

4.2. Parameters setting

In this part, we show how to choose parameters for the studied algorithms. Firstly, for the common parameters μ and γ of classical ADMM, GADMM, IADMM_CCMY and Algorithm 1, we take their values as $1/\sqrt{m}$ and 0.01, respectively. Secondly, for the private parameters of the Algorithm 1, we fix $\sigma = 0.01$, δ and relaxation parameter λ_k are

$$\delta = 1 + \frac{\alpha^2(1+\alpha) + \alpha\sigma}{1-\alpha^2} \text{ and } \lambda_k = 2\frac{\delta - \alpha[\alpha(1+\alpha) + \alpha\delta + \sigma]}{\delta[1 + \alpha(1+\alpha) + \alpha\delta + \sigma]}, \tag{4.3}$$

where α takes four different values 0.05, 0.1, 0.2 and 0.3, respectively, and let inertial parameter $\alpha_k = \alpha$. The relaxation parameter λ_k of GADMM is constant 1.6 and the inertial parameter α_k

of IADMM_CCMY are the same as that of Algorithm 1.

For the convenience of subsequent experiments, we need to find the optimal inertial parameter α_k of the two inertial algorithms of IADMM_CCMY and Algorithm 1. The different selection of inertial parameters are listed in Table 1. In the experiment, the order of objective matrix b is $m = 1000$, the rank of low rank matrix u^* is $r = 0.1m$, and the sparsity of sparse matrix v^* is $\|v^*\|_0 = 0.05m^2$, respectively.

Methods	γ	σ	Inertial parameter α_k	Relaxation parameter λ_k
Algorithm 1-1	0.01	0.01	0.05	1.7874
Algorithm 1-2			0.1	1.6019
Algorithm 1-3			0.2	1.2496
Algorithm 1-4			0.3	0.9243
IADMM_CCMY-1		None	0.05	None
IADMM_CCMY-2			0.1	None
IADMM_CCMY-3			0.2	None
IADMM_CCMY-4			0.3	None

Table 1 Parameters selection of the IADMM_CCMY and Algorithm 1

We define the relative error $\mathrm{rel}\, u$, $\mathrm{rel}\, v$ and $\mathrm{rel}\, b$ as the stopping criterion, i.e.,

$$\mathrm{rel}\, u := \frac{\|u^{k+1} - u^k\|_F}{\|u^k\|_F}, \ \mathrm{rel}\, v := \frac{\|v^{k+1} - v^k\|_F}{\|v^k\|_F}, \ \mathrm{rel}\, b := \frac{\|b^{k+1} - b^k\|_F}{\|b^k\|_F},$$
$$\max(\mathrm{rel}\, u, \mathrm{rel}\, v, \mathrm{rel}\, b) \le \varepsilon,$$

where ε is a given small constant.

Methods	$\varepsilon = 1e-5$				$\varepsilon = 1e-7$			
	k	$\mathrm{rel}\, u^*$	$\mathrm{rel}\, v^*$	$\mathrm{rank}(u^*)$	k	$\mathrm{rel}\, u^*$	$\mathrm{rel}\, v^*$	$\mathrm{rank}(u^*)$
Algorithm 1-1	70	5.3242e-6	1.2665e-6	100	101	1.7586e-6	6.0458e-7	100
Algorithm 1-2	56	6.8826e-6	1.8277e-6	100	96	1.7588e-6	6.0457e-7	100
Algorithm 1-3	49	1.3062e-5	4.1950e-6	100	69	4.8588e-6	1.7194e-6	100
Algorithm 1-4	53	1.7610e-5	5.6121e-6	100	77	4.8727e-6	1.7199e-6	100
IADMM_CCMY-1	55	3.1465e-5	9.6907e-6	100	91	4.8607e-6	1.7197e-6	100
IADMM_CCMY-2	53	2.9038e-5	9.6133e-6	100	87	4.8698e-6	1.7199e-6	100
IADMM_CCMY-3	49	2.8558e-5	9.1194e-6	100	78	4.8658e-6	1.7196e-6	100
IADMM_CCMY-4	50	1.7318e-5	5.6383e-6	100	72	4.8530e-6	1.7192e-6	100

Table 2 Numerical experimental results of IADMM_CCMY and Algorithm 1 under different inertial parameters α_k ($\mathrm{rel}\, u^*$ and $\mathrm{rel}\, v^*$ are defined as $\frac{\|u^k - u^*\|_F}{\|u^*\|_F}$ and $\frac{\|v^k - v^*\|_F}{\|v^*\|_F}$, respectively)

From Table 2, we can see that when the inertial parameters α_k of IADMM_CCMY and Algorithm 1 are 0.3 and 0.2, respectively, the experimental results are the best. In the following experiments, we fixed the inertial parameters α_k of IADMM_CCMY and Algorithm 1 to 0.3 and 0.2, respectively.

4.3. Results and discussions

In the experiment, the order of objective matrix b is $m = 500$, $m = 800$ and $m = 1000$, the rank of low rank matrix u^* is $r = 0.05m$ and $r = 0.1m$, and the sparsity of sparse matrix v^* is $\|v^*\|_0 = 0.05m^2$ and $\|v^*\|_0 = 0.1m^2$, respectively.

	m	Methods	k	$\mathrm{rel}\, u^*$	$\mathrm{rel}\, v^*$	$\mathrm{rank}(u^k)$
$\mathrm{rank}(u^*) = 0.05m$ $\lVert v^*\rVert_0 = 0.05m^2$ $\varepsilon = 1e-7$	500	ADMM	58	1.6323e-5	3.6376e-6	25
		GADMM	45	1.6199e-5	3.6358e-6	25
		IADMM_CCMY	46	1.6150e-5	3.6351e-6	25
		Algorithm 1	48	1.6153e-5	3.6351e-6	25
	800	ADMM	89	5.3299e-6	1.5083e-6	40
		GADMM	76	2.3394e-6	6.5742e-7	40
		IADMM_CCMY	68	5.3209e-6	1.5081e-6	40
		Algorithm 1	63	5.3177e-6	1.5081e-6	40
	1000	ADMM	76	6.3916e-6	2.0164e-6	50
		GADMM	57	6.3510e-6	2.0155e-6	50
		IADMM_CCMY	57	6.3514e-6	2.0155e-6	50
		Algorithm 1	55	6.3540e-6	2.0156e-6	50
$\mathrm{rank}(u^*) = 0.05m$ $\lVert v^*\rVert_0 = 0.1m^2$ $\varepsilon = 1e-7$	500	ADMM	89	1.7912e-5	2.8058e-6	25
		GADMM	62	1.7661e-5	2.8031e-6	25
		IADMM_CCMY	68	1.7626e-5	2.8027e-6	25
		Algorithm 1	64	1.7658e-5	2.8031e-6	25
	800	ADMM	118	8.9767e-6	1.7795e-6	40
		GADMM	90	4.9539e-6	9.9398e-7	40
		IADMM_CCMY	94	4.9658e-6	9.9416e-7	40
		Algorithm 1	92	4.9539e-6	9.9398e-7	40
	1000	ADMM	127	5.6708e-6	1.1938e-6	50
		GADMM	109	3.4869e-6	7.5976e-7	50
		IADMM_CCMY	101	4.4912e-6	9.8826e-7	50
		Algorithm 1	98	4.4909e-6	9.8826e-7	50
$\mathrm{rank}(u^*) = 0.1m$ $\lVert v^*\rVert_0 = 0.05m^2$ $\varepsilon = 1e-7$	500	ADMM	68	7.8951e-6	1.7185e-6	50
		GADMM	49	7.8838e-6	1.7181e-6	50
		IADMM_CCMY	54	7.8842e-6	1.7182e-6	50
		Algorithm 1	51	7.8840e-6	1.7181e-6	50
	800	ADMM	78	7.4927e-6	2.1961e-6	80
		GADMM	60	4.3186e-6	1.2655e-6	80
		IADMM_CCMY	68	4.3188e-6	1.2655e-6	80
		Algorithm 1	64	4.3262e-6	1.2657e-6	80
	1000	ADMM	95	5.4043e-6	1.8768e-6	100
		GADMM	69	4.8821e-6	1.7201e-6	100
		IADMM_CCMY	72	4.8530e-6	1.7192e-6	100
		Algorithm 1	69	4.8588e-6	1.7194e-6	100
$\mathrm{rank}(u^*) = 0.1m$ $\lVert v^*\rVert_0 = 0.1m^2$ $\varepsilon = 1e-7$	500	ADMM	104	8.1931e-6	1.2617e-6	50
		GADMM	84	6.3919e-6	9.9343e-7	50
		IADMM_CCMY	86	6.3780e-6	9.9311e-7	50
		Algorithm 1	88	6.3918e-6	9.9343e-7	50
	800	ADMM	138	7.3081e-6	1.4414e-6	80
		GADMM	103	2.1651e-6	4.6292e-7	80
		IADMM_CCMY	113	2.1650e-6	4.6292e-7	80
		Algorithm 1	107	2.1652e-6	4.6292e-7	80
	1000	ADMM	140	5.0963e-6	1.1111e-6	100
		GADMM	103	1.8990e-6	4.3722e-7	100
		IADMM_CCMY	99	5.0276e-6	1.1086e-6	100
		Algorithm 1	104	1.9001e-6	4.3725e-7	100

Table 3 Comparison of numerical experimental results of ADMM, GADMM, IADMM_CCMY and Algorithm 1 ($\mathrm{rel}\, u^*$ and $\mathrm{rel}\, v^*$ are defined as $\frac{\lVert u^k-u^*\rVert_F}{\lVert u^*\rVert_F}$ and $\frac{\lVert v^k-v^*\rVert_F}{\lVert v^*\rVert_F}$, respectively)

Table 3 is a comparison of the numerical experimental results of the classical ADMM, GADMM, IADMM_CCMY and Algorithm 1. We conclude from Table 3 that the two inertial ADMMs (IADMM_CCMY and Algorithm 1) and the GADMM are better than the classical ADMM in terms of iteration numbers and accuracy. The IADMM_CCMY and Algorithm 1 are similar in terms of accuracy. The proposed Algorithm 1 is better than IADMM_CCMY in the number of iterations in most cases. Besides, the proposed Algorithm 1 is also comparable with the GADMM (1.5). In some cases, the number of iteration of the GADMM is higher than that of Algorithm 1. For example, when $m = 800$, $\mathrm{rank}(u^*) = 0.05m$ and $\|v^*\|_0 = 0.05m^2$, and $m = 1000$, $\mathrm{rank}(u^*) = 0.05m$ and $\|v^*\|_0 = 0.1m^2$. But, in most cases, the number of iteration of the GADMM is less than Algorithm 1.

5. Conclusions

ADMM is a popular method for solving many structural convex optimization problems. In this paper, we proposed an inertial ADMM for solving the two-block separable convex optimization problem with linear equality constraints (1.1), which derived from the inertial Douglas-Rachford splitting algorithm applied to the dual of (1.1). The obtained algorithm generalized the inertial ADMM of [23]. Furthermore, we proved the convergence results of the proposed algorithm under mild conditions on the parameters. Numerical experiments for solving the R-PCA (4.2) showed the advantage of the proposed algorithm over existing iterative algorithms including the classical ADMM and the inertial ADMM introduced by Chen et al. [21]. We also found the proposed algorithm is comparable with the GADMM (1.5).

Acknowledgements The authors would like to thank the two anonymous reviewers for their suggestions and comments to improve the manuscript. The authors would also like to thank the editors for their hard work.

References

[1] R. GLOWINSKI, A. MARROCO. *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires*. Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér., 1975, **9**(R-2): 41–76. (in French)

[2] D. GABAY, B. MERCIER. *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*. Comput. Math. Appl., 1976, **2**: 17–40.

[3] J. ECKSTEIN, D. BERTSEKAS. *On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators*. Math. Program., 1992, **55**(1): 293–318.

[4] T. GOLDSTEIN, S. OSHER. *The split bregman method for ℓ_1-regularized problems*. SIAM J Imaging Sci., 2009, **2**(2): 1–21.

[5] E. ESSER. *Applications of lagrangian-based alternating direction methods and connections to split bregman*. CAM Report 09-31, UCLA, April 2009.

[6] S. BOYD, N. PARIKH, E. CHU, et al. *Distrituted optimization and statistical learning via the alternating direction method of multipliers*. Found. Trends Mach. Learn., 2010, **3**: 1–122.

[7] Bingsheng HE, Xiaoming YUAN. *On the $o(1/n)$ convergence rate of the douglas-rachford alternating direction method*. SIAM J. Numer. Anal., 2012, **50**(2): 700–709.

[8] R. D. MONTEIRO, B. F. SVAITER. *Iteration-complexity of block-decomposition algoirhms and the alternating direction method of multipliers*. SIAM J. Optim., 2013, **23**(1): 475–507.

[9] E. X. FANG, Bingsheng HE, Han LIU, et al. *Generalized alternating direction method of multipliers: new theoretical insights and applications*. Math. Program. Comput., 2015, **7**(2): 149–187.

[10] Bingsheng HE, Xiaoming YUAN. *On non-ergodic convergence rate of douglas-rachford alternating direction method of multipliers*. Numer. Math., 2015, **130**(3): 567–577.

[11] Bingsheng HE, Liusheng HOU, Xiaoming YUAN. *On full jacobian decomposition of the augmented lagrangian method for separable convex programming*. SIAM J. Optim., 2015, **25**(4): 2274–2312.

[12] Bingsheng HE, Min TAO, Xiaoming YUAN. *A splitting method for separable convex programming*. IMA J. Numer. Anal., 2015, **35**: 394–426.

[13] Caihua CHEN, Bingsheng HE, Yinyu YE, et al. *The direct extension of admm for multi-block minimization problems is not necessarily convergent*. Math. Program., 2016, **155**(1-2): 57–79.

[14] Xingju CAI, Guoyong GU, Bingsheng HE, et al. *A proximal point algorithm revisit on the alternating direction method of multipliers*. Sci. China Math., 2013, **56**(10): 2179–2186.

[15] F. ALVAREZ, H. ATTOUCH. *An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping*. Set-Valued Anal., 2001, **9**(1-2): 3–11.

[16] H. ATTOUCH, A. CABOT. *Convergence of a relaxed inertial proximal algorithm for maximally monotone operators*. Math. Program., 2019, 1–45.

[17] H. ATTOUCH, A. CABOT. *Convergence rates of inertial forward-backward algorithms*. SIAM J. Optim., 2018, **28**(1): 849–874.

[18] H. ATTOUCH, A. CABOT. *Convergence of a relaxed inertial forward-backward algorithm for structured monotone inclusions*. Appl. Math. Optim., 2019, **80**(3): 547–598.

[19] R. I. BOŢ, E. R. CSETNEK. *An inertial forward-backward-forward primal-dual splitting algorithm for solving monotone inclusion problems*. Numer. Algorithms, 2016, **71**(3): 519–540.

[20] Fuying CUI, Yuchao TANG, Yang YANG. *An inertial three-operator splitting algorithm with applications to image inpainting*. Appl. Set-Valued Anal. Optim., 2019, **1**(2): 113–134.

[21] Caihua CHEN, R. H. CHAN, Shiqian MA, et al. *Inertial proximal admm for linearly constrained separable convex optimization*. SIAM J. Imaging Sci., 2015, **8**(4): 2239–2267.

[22] H. ATTOUCH, M. SOUEYCATT. *Augmented lagrangian and proximal alternating direction methods of multipliers in hilbert spaces, applications to games, pde's and control*. Pacific. J. Optim., 2008, **5**(1): 17–37.

[23] R. I. BOŢ, E. R. CSETNEK. *An inertial alternating direction method of multipliers*. Minimax Theory Appl., 2016, **1**(1): 29–49.

[24] R. I. BOŢ, E. R. CSETNEK, C. HENDRICH. *Inertial douglas-rachford splitting for monotone inclusion problems*. Appl. Math. Comput., 2015, **256**: 472–487.

[25] H. H. BAUSCHKE, P. L. COMBETTES. *Convex Analysis and Motonone Operator Theory in Hilbert Spaces*. Springer, London, Second Edition, 2017.

[26] Xiaoming YUAN, Junfeng YANG. *Sparse and low-rank matrix decomposition via alternating direction methods*. Pacific J. Optim., 2013, **9**(1): 167–180.

[27] E. J. CANDÈS, X. LI, Y. MA, et al. *Robust principal component analysis?* J. ACM, 2009, **58**(1): 1–37.

[28] T. BOUWMANS, E. H. ZAHZAH. *Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance*. Comput. Vis. Image Under., 2014, **122**: 22–34.

[29] Shiqian MA, N. S. AYBAT. *Efficient optimization algoirhms for robust principal component analysis and its variants*. Procedings of the IEEE, 2018, **106**(8): 1411–1426.

[30] V. CHANDRASEKARAN, S. SANGHAVI, P. A. PARRILO, et al. *Rank-sparsity incoherence for matrix decompositions*. SIAM J. Optim., 2011, **21**(2): 572–596.