

An adaptive linearized alternating direction multiplier method with a relaxation step for convex programming

Boran Wang

College of Science, Minzu University of China, Beijing, 100081, China.

Contributing authors: 1932210936@qq.com;

Abstract

Alternating direction multiplication is a powerful technique for solving convex optimisation problems. When challenging subproblems are encountered in the real world, it is useful to solve them by introducing neighbourhood terms. When the neighbourhood matrix is positive definite, the algorithm converges but at the same time makes the iteration step small. Recent studies have revealed the potential non-positive definiteness of the neighbourhood matrix. In this paper, we present an adaptive linearized alternating direction multiplier method with a relaxation step, combining the relaxation step with an adaptive technique. The novelty of the method is to use the information of the current iteration point to dynamically select the neighbourhood matrix, increase the iteration step size, and speed up the convergence of the algorithm. We prove the global convergence of the algorithm theoretically and illustrate the effectiveness of the algorithm using numerical experiments.

Keywords: variational inequality, an adaptive linearized ADMM , a relaxation step , global convergence.

1 Introduction

This paper considers the following separable convex optimization problem with linear constraints:

$$\min \{f(x) + g(y) \mid Ax + By = b, x \in X, y \in Y\} \quad (1)$$

where $f : R^{n_1} \rightarrow R$ and $g : R^{n_2} \rightarrow R$ are convex functions (but not necessarily smooth), $A \in R^{m \times n_1}$, $B \in R^{m \times n_2}$, $b \in R^m$, $X \subseteq R^{n_1}$ and $Y \subseteq R^{n_2}$ are closed convex sets. Problem (1) has recently been widely used in several areas such as image processing[1, 2], statistical learning[3] and communication networks[4]. There exist numerous effective methods for addressing problem (1), including the primal-dual method, the augmented Lagrange method, and the alternating direction multiplier method. Notably, the alternating direction multiplier method has garnered significant attention in recent years due to its simplicity and efficiency. Many scholars have introduced several effective variations of this algorithm tailored to practical challenges.

The alternate direction multiplier approach, which was first put forth by Gabay and Mercier[5] and Glowinski and Marrocco[6], is widely recognized as an effective way to solve (1). For the progress of research on the alternating direction multiplier method, we refer to[7]. The method possesses fast convergence speed and strong convergence performance, making it a crucial tool for addressing convex optimization problems with divisibility. It can decompose a large problem into two or more small-scale subproblems and then iteratively solve each subproblem to enhance solution speed. Its direct, adaptable, and practical nature is particularly effective in tackling convex optimization problems with separability. The alternating direction multiplier method can be described as follows:

$$\begin{cases} x^{k+1} = \arg \min_x \{L_\beta(x, y^k, \lambda^k) | x \in X\} \\ y^{k+1} = \arg \min_y \{L_\beta(x^{k+1}, y, \lambda^k) | y \in Y\} \\ \lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) \end{cases} \quad (2)$$

where $L_\beta(x, y, \lambda)$ is the augmented Lagrangian function of (1)

$$L_\beta(x, y, \lambda) = f(x) + g(y) - \lambda^T (Ax + By - b) + \frac{\beta}{2} \|Ax + By - b\|_2^2.$$

where $\lambda \in R^m$ represents the Lagrange multiplier, and $\beta > 0$ represents the penalty parameter. For simplicity, the penalty parameter β is fixed in our discussion.

A significant focus in the diverse research literature concerning the alternating direction multiplier method is the exploration of efficient strategies for solving its subproblems. In certain scenarios, the functions $f(x)$ and $g(y)$, or the coefficient matrices A and B may exhibit unique properties and structures. Leveraging these characteristics, researchers can extend the framework presented in (2) to devise algorithms tailored to specific applications while upholding theoretical convergence guarantees. This principle underscores the critical role of effectively applying the alternating direction multiplier method across various fields and applications. To delve deeper into this concept, let's examine the y -subproblem outlined in (2). As previously noted, the function $g(y)$, the matrix B , and the set Y are pivotal in addressing the y -subproblems. Typically, when these the function, matrix, and set are in a generic form, the solution to (2) can be obtained using straightforward iterative techniques, we refer to [8–10]. However, in practical scenarios, the functions, matrices, and sets Y associated with the y -subproblem may possess distinct characteristics, necessitating more efficient solution methods. When the matrix B is not an identity matrix, the y -subproblem (2) can

be reformulated as follows:

$$y^{k+1} = \arg \min_y \left\{ g(y) + \frac{\beta}{2} \left\| By + \left(Ax^{k+1} - b - \frac{1}{\beta} \lambda^k \right) \right\|^2 \mid y \in Y \right\} \quad (3)$$

After linearizing the quadratic term $\left\| By + \left(Ax^{k+1} - b - \frac{1}{\beta} \lambda^k \right) \right\|^2$ in (3), it is obtained:

$$y^{k+1} = \arg \min_y \left\{ g(y) + \frac{r}{2} \left\| y - \left(y^k + \frac{1}{r} q^k \right) \right\|^2 \mid y \in Y \right\}. \quad (4)$$

where

$$q^k = B^T (\lambda^k - \beta (Ax^{k+1} + By^k - b)). \quad (5)$$

where $r > 0$ is a constant, and thus we obtain the linearized alternating direction multiplier method[11]:

$$\begin{cases} x^{k+1} = \arg \min_x \{ L_\beta(x, y^k, \lambda^k) \mid x \in X \} \\ y^{k+1} = \arg \min_y \left\{ L_\beta(x^{k+1}, y, \lambda^k) + \frac{r}{2} \left\| y - \left(y^k + \frac{1}{r} q^k \right) \right\|^2 \mid y \in Y \right\} \\ \lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) \end{cases} \quad (6)$$

Linearized alternating direction multiplier method is widely used with compressed perception[12], and image processing[13, 14], etc. .

A more general alternating direction multiplier method[15] can be written as follows:

$$\begin{cases} x^{k+1} = \arg \min_x \{ L_\beta(x, y^k, \lambda^k) \mid x \in X \} \\ y^{k+1} = \arg \min_y \left\{ L_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2} \|y - y^k\|_D^2 \mid y \in Y \right\} \\ \lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) \end{cases} \quad (7)$$

where $D \in R^{n_2 \times n_2}$ is positive definite, and thus, the linearized alternating direction multiplier method can be viewed as a special case of the general alternating direction multiplier method, where the positive definite proximal term.

$$D = rI_{n_2} - \beta B^T B \quad \text{and} \quad r > \beta \|B^T B\| \quad (8)$$

Since in many practical applications, only one subproblem in (6) needs to be linearized, in this paper, we only need to consider the case of linearizing the y -subproblem in (7).

In order to improve the applicability of the linearized alternating direction multiplier method, a number of improved linearized alternating direction multiplier methods[16–20] have been proposed by some scholars.

Literature[21] proposes an indefinite proximity alternating direction multiplier method with the following iteration format:

$$\begin{cases} x^{k+1} = \arg \min_x \{L_\beta(x, y^k, \lambda^k) | x \in X\} \\ y^{k+1} = \arg \min_y \left\{ L_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2} \|y - y^k\|_{D_0}^2 | y \in Y \right\} \\ \lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) \end{cases} \quad (9)$$

where

$$D_0 = \tau r I_{n_2} - \beta B^T B, \quad 0.8 \leq \tau < 1 \quad \text{and} \quad r > \beta \|B^T B\|.$$

He et al. proposed an optimal linearized ADMM in the literature [22] with the iterative steps:

$$\begin{cases} x^{k+1} = \arg \min_x \{L_\beta(x, y^k, \lambda^k) | x \in X\} \\ y^{k+1} = \arg \min_y \left\{ L_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2} \|y - y^k\|_D^2 | y \in Y \right\} \\ \lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) \end{cases} \quad (10)$$

where

$$D = \tau r I - \beta B^T B, \quad 0.75 < \tau < 1 \quad \text{and} \quad \rho > \beta \|B^T B\|$$

It is easy to see that the matrix D is not necessarily semipositive definite. In the paper the authors prove that 0.75 is an optimal lower bound for τ .

Another important issue for ADMM is to design an accelerated version of the original ADMM by slightly modifying it through a simple relaxation scheme. Note that ADMM is closely related to Proximity Point Algorithms[23] (PPA). For the Proximity Point Algorithm, the convergence rate is improved if an additional over-relaxation step is added to the basic variables; see Tao[24] for a relaxation variant of the PPA and proof of its linear convergence result. Gu and Yang[25] further proved the optimal linear convergence rate of the relaxation PPA under regularity conditions. As Boyd et al. proved in[3], the execution of ADMM relies on the inputs of (y^k, λ^k) and does not require x^k at all. Thus, x plays the role of an intermediate variable in (2), while (y^k, λ^k) is the basic variable. It is therefore natural to ask whether it is possible to include a relaxation step for the basic variables (y^k, λ^k) in the ADMM scheme (2) to obtain a faster ADMM-type method. This idea leads to:

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ f(x) - x^T A^T \lambda^k + \frac{\beta}{2} \|Ax + By^k - b\|^2 \right\} \\ \hat{y}^k = \arg \min_y \left\{ g(y) - y^T B^T \lambda^k + \frac{\beta}{2} \|Ax^{k+1} + By - b\|^2 \right\} \\ \hat{\lambda}^k = \lambda^k - \beta (Ax^{k+1} + B\hat{y}^k - b) \end{cases} \quad (11)$$

$$\begin{pmatrix} y^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ \lambda^k \end{pmatrix} - \sigma \begin{pmatrix} y^k - \hat{y}^k \\ \lambda^k - \hat{\lambda}^k \end{pmatrix} \quad (12)$$

where the relaxation factor $\sigma \in (0, 2)$. Based on the above discussion, In this study, we introduce an adaptive linearized alternating direction multiplier method with a relaxation step that incorporates both a relaxation step and an adaptive technique. Our approach adopts a unified framework of variational inequalities and establishes the global convergence of this adaptive linearized alternating direction multiplier method with a relaxation step through theoretical analysis. Through the resolution of the Lasso problem, we demonstrate the algorithm's outstanding numerical performance. These findings will propel advancements in optimization algorithms and offer more efficient tools and methodologies for addressing practical challenges.

The rest of the paper is organized as follows. Section 2 gives some preliminaries; Section 3 gives iterative steps and proof of convergence of the algorithm; Section 4 gives the numerical experiments; and Section 5 draws the conclusions.

2 Preliminaries

We present some preliminaries that will be used in convergence analysis. In this article, the symbol $\|\cdot\|$ denotes the two-norm $\|\cdot\|_2$. $\|x\|_D^2 := x^T D x$ is the matrix norm, where $D \in R^{n \times n}$ is a symmetric positive definite matrix and the vector $x \in R^n$. When D is not a positive definite matrix, we still use the above notation.

2.1 characterization of variational inequalities

Define the Lagrangian function corresponding to problem (1) as:

$$L(x, y, \lambda) = f(x) + g(y) - \lambda^T (Ax + By - b) \quad (13)$$

In (13), (x, y) and λ denote primitive and dual variables, respectively. If $(x^*, y^*, \lambda^*) \in X \times Y \times R^m$ satisfies the following inequality:

$$L(x^*, y^*, \lambda) \leq L(x^*, y^*, \lambda^*) \leq L(x, y, \lambda^*), \quad \forall (x, y, \lambda) \in X \times Y \times R^m = \Omega.$$

then (x^*, y^*, λ^*) is called a saddle point of $L(x, y, \lambda)$.

The above inequality is equivalent to the following variational inequality form:

$$\begin{cases} x^* \in X, & f(x) - f(x^*) + (x - x^*)^T (-A^T \lambda^*) \geq 0, \quad \forall x \in X. \\ y^* \in Y, & g(y) - g(y^*) + (y - y^*)^T (-A^T \lambda^*) \geq 0, \quad \forall y \in Y. \\ \lambda^* \in R^m, & (\lambda - \lambda^*)^T (Ax^* + By^* - b) \geq 0, \quad \forall \lambda \in R^m. \end{cases} \quad (14)$$

The above variational inequality can be written in the following form:

$$VI(\Omega, F, \theta) : \quad w^* \in \Omega, \quad \theta(w) - \theta(w^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega. \quad (15)$$

where

$$\theta(u) = f(x) + g(y), u = \begin{pmatrix} x \\ y \end{pmatrix}, v = \begin{pmatrix} y \\ \lambda \end{pmatrix}, w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, F(w) = \begin{pmatrix} -A^T \lambda \\ -B^T \lambda \\ Ax + By - b \end{pmatrix} \quad (16)$$

Owing to:

$$(w_1 - w_2)^T (F(w_1) - F(w_2)) \geq 0, \quad \forall w_1, w_2 \in \Omega.$$

So, F is a monotone operator.

The problem (1) is reformulated as the variational inequality (15) in this manner. We denote the set of solutions of the variational inequality $VI(\Omega, F, \theta)$ as Ω^* , where Ω^* represents the set of non-empty solutions.

2.2 some notation

For the convenience of the proof, first define some auxiliary variables and matrices. Let

$$Q_{k+1} = \begin{pmatrix} \tau_k r \beta I_{n_2} & O \\ -B & \frac{1}{\beta} I_m \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} \sigma I_{n_2} & O \\ -\sigma \beta B & \sigma I_m \end{pmatrix}. \quad (17)$$

$$H_{k+1} = \frac{1}{\sigma} \begin{pmatrix} \tau_k r \beta I_{n_2} & O \\ O & \frac{1}{\beta} I_m \end{pmatrix}. \quad (18)$$

$$\tilde{w}^k = \begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} x^{k+1} \\ \hat{y}^k \\ \lambda^k - \beta (Ax^{k+1} + By^k - b) \end{pmatrix}. \quad (19)$$

$$\tilde{v}^k = \begin{pmatrix} \tilde{y}^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} \hat{y}^k \\ \lambda^k - \beta (Ax^{k+1} + By^k - b) \end{pmatrix} \quad (20)$$

Lemma 2.1. The Q_{k+1} , H_{k+1} and M defined in (17) and (18) satisfy:

$$Q_{k+1} = H_{k+1} M \quad \text{and} \quad H_{k+1} \succeq 0, \quad (21)$$

Proof: (21) clearly hold.

Lemma 2.2. (Robbins-Siegmund Theorem[26]) a^k, b^k, c^k and d^k are non-negative sequences and there are :

$$a^{k+1} \leq (1 + b^k) a^k + c^k - d^k, \quad \forall k = 0, 1, 2, \dots \quad (22)$$

if $\sum_{k=0}^{+\infty} b^k < +\infty$ and $\sum_{k=0}^{+\infty} c^k < +\infty$, so $\lim_{k \rightarrow \infty} a^k$ exists and is bounded, while there are $\sum_{k=0}^{+\infty} d^k < +\infty$.

Definition 2.1. If a function $f(\cdot)$ is a nonsmooth convex function on a convex set, if the following inequality holds:

$$f(u) \geq f(v) + \phi^T(u - v), \quad \forall u \in \Omega.$$

then ϕ is said to be the subgradient of the function $f(\cdot)$ at $v \in \Omega$; The set consisting of all subgradients is called the subdifferential of the function $f(\cdot)$ at $v \in \Omega$, denoted $\partial f(v)$.

2.3 Stopping criterion

In the context of the two-block divisible convex optimization problem (1), we investigate the optimality conditions for each subproblem during the iterative process of the Alternating Direction Method of Multipliers .

According to the optimality condition theorem, if x^* , y^* are optimal solutions to the convex optimisation problem (1) and λ^* is the corresponding Lagrange multiplier, then the following conditions are satisfied.

$$0 \in \partial f(x^*) - A^T \lambda^* \quad (23)$$

$$0 \in \partial g(y^*) - B^T \lambda^* \quad (24)$$

$$Ax^* + By^* = b \quad (25)$$

In this context, condition (25) is denoted as the original feasibility condition, while conditions (23) and (24) are labeled as the dual feasibility conditions.

According to the optimality conditions for the y -subproblem in (1), we have:

$$\begin{aligned} 0 &\in \partial g(y^{k+1}) - B^T \lambda^k + \beta B^T (Ax^{k+1} + By^{k+1} - b) \\ &= \partial g(y^{k+1}) - B^T \lambda^{k+1}. \end{aligned} \quad (26)$$

According to the optimality conditions for the x -subproblem in (1), we have:

$$0 \in \partial f(x^{k+1}) - A^T \lambda^k + \beta A^T (Ax^{k+1} + By^k - b). \quad (27)$$

From the definition of λ^{k+1} in (1), the above equation can be equated to:

$$\begin{aligned} 0 &\in \partial f(x^{k+1}) - A^T (\lambda^k - \beta (Ax^{k+1} + By^{k+1} - b) - \beta B (y^k - y^{k+1})) \\ &= \partial f(x^{k+1}) - A^T \lambda^{k+1} + \beta A^T B (y^k - y^{k+1}). \end{aligned} \quad (28)$$

(28) is equivalent to:

$$\beta A^T B (y^{k+1} - y^k) \in \partial f(x^{k+1}) - A^T \lambda^{k+1}. \quad (29)$$

When comparing (29) with condition (23), it is evident that the additional term is $\beta A^T B (y^{k+1} - y^k)$. Thus, to verify dual feasibility, it is adequate to examine the residual $\beta A^T B (y^{k+1} - y^k)$.

In summary, to ascertain the convergence of the alternating direction multiplier method, it is necessary to verify if the two residuals p^{k+1} and d^{k+1} are sufficiently

small.
where

$$\begin{aligned} p^{k+1} &= \|Ax^{k+1} + By^{k+1} - b\|. \\ d^k &= \|\beta A^T B (y^{k+1} - y^k)\|. \end{aligned}$$

3 Algorithm and convergence analysis

3.1 New algorithms

Suppose that $f : R^n \rightarrow R \cup \{+\infty\}$ and $g : R^n \rightarrow R \cup \{+\infty\}$ are proper, closed, convex functions.

The main iterative steps of the algorithm are:

Algorithm 1. An adaptive linearized alternating direction multiplier method with a relaxation step .

Set up: $\Omega = X \times Y \times R^m, \beta > 0, \tau_k > 0, p^k \geq 0, d^k \geq 0, r = \|B^T B\|, \sigma \in (0, 2), \varepsilon \in (0, 2 - \sigma), \Upsilon > 1, \rho > 1, \epsilon^{pri} > 0, \epsilon^{dual} > 0$. choose parameter sequence $\{\eta_k\}$ and $\{s_k\}$, where $\sum_{k=0}^{\infty} \eta_k < +\infty$ and $\sum_{k=0}^{\infty} s_k < +\infty$.

Step 0. Input: $w^0 = (x^0, y^0, \lambda^0) \in \Omega, \beta, \sigma, \tau_0, \tau_{min}, p^0, d^0, \Upsilon, \varepsilon, r, \rho, \epsilon^{pri}, \epsilon^{dual}$. set up $k = 0$.

Step 1. Calculate: $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in X \times Y \times R^m$.

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ f(x) - (\lambda^k)^T Ax + \frac{\beta}{2} \|Ax + By^k - b\|^2 \right\} \\ \hat{y}^k = \arg \min_y \left\{ g(y) - (\lambda^k)^T By + \frac{\beta}{2} \|Ax^{k+1} + By - b\|^2 \right. \\ \quad \left. + \frac{1}{2} \|y - y^k\|_{D_k}^2 \right\} \\ \hat{\lambda}^k = \lambda^k - \beta (Ax^{k+1} + B\hat{y}^k - b) \end{cases} \quad (30)$$

where $D_k = \tau_k r I_{n_2} - \beta B^T B$.

$$\begin{cases} y^{k+1} = y^k - \sigma (y^k - \hat{y}^k) \\ \lambda^{k+1} = \lambda^k - \sigma (\lambda^k - \hat{\lambda}^k) \end{cases} \quad (31)$$

where $\sigma \in (0, 2)$.

Step 2. If any one of the following conditions holds:

$$\text{Condition 1. } \Theta_1^k > \Theta_2^k. \quad (32)$$

$$\text{Condition 2. } y^{k+1} = y^k.$$

where $\Theta_1^k = (2 - \sigma) \tau_k r \|y^k - y^{k+1}\|^2, \Theta_2^k = \frac{1}{\varepsilon} \|B (y^k - y^{k+1})\|^2$.
then go to step 3. otherwise $\tau_k = \gamma * \tau_k (\gamma > 1)$, turn to step 1.

Step 3. If $\Theta_1^k - \Theta_2^k \geq \Upsilon \Theta_2^k$. then set $t_{k+1} = \max \left\{ \frac{\tau_k}{1 + \eta_{k+1}}, \tau_{min} \right\}$, Otherwise $t_{k+1} = \tau_k$.

Step 4. If any one of the following conditions holds:

$$\text{Condition1. } p^{k+1} > (1 + s_k) p^k.$$

$$\text{Condition2. } d^{k+1} > (1 + s_k) d^k.$$

where $p^{k+1} = \|Ax^{k+1} + By^{k+1} - b\|$, $d^{k+1} = \|\beta A^T B (y^{k+1} - y^k)\|$.

then set $\tau_{k+1} = \rho * t_{k+1}$ ($\rho > 1$), Otherwise $\tau_{k+1} = t_{k+1}$.

Step 5. If the stopping criterion is satisfied, return to step 6.

where, the stopping criterion is:

$$\|Ax^{k+1} + By^{k+1} - b\| \leq \epsilon^{pri} \quad \text{and} \quad \|\beta A^T B (y^{k+1} - y^k)\| \leq \epsilon^{dual} \quad (33)$$

Otherwise make $k = k + 1$, and return to step 1.

Step 6. Output: $x^{k+1}, y^{k+1}, f(x^{k+1}) + g(y^{k+1})$.

3.2 Global convergence analysis

In this section, we prove the global convergence for the proposed method. Before proceeding, we need the following lemma.

Lemma 3.1. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1, and $\{\tilde{w}^k\}$ be defined in (19), then, for any $\tilde{w}^k \in \Omega$, there are:

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T Q_{k+1} (v^k - \tilde{v}^k), \quad \forall w \in \Omega. \quad (34)$$

Proof: by the optimality condition for the x subproblem in Algorithm 1: for any $\tilde{x}^k \in X$, there are:

$$f(x) - f(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T \lambda^k + \beta A^T (A\tilde{x}^k + By^k - b)\} \geq 0, \quad \forall x \in X.$$

Since $\tilde{\lambda}^k = \lambda^k - \beta (A\tilde{x}^k + By^k - b)$ in (19), then:

$$f(x) - f(\tilde{x}^k) + (x - \tilde{x}^k)^T (-A^T \tilde{\lambda}^k) \geq 0. \quad \forall x \in X. \quad (35)$$

From the optimality conditions for the y -subproblem in Algorithm 1, it follows that for any $\tilde{y}^k \in Y$, there are:

$$\begin{aligned} g(y) - g(\tilde{y}^k) + (y - \tilde{y}^k)^T \{-B^T \lambda^k + \beta B^T (A\tilde{x}^k + B\tilde{y}^k - b) \\ + (\tau_k r I_{n_2} - \beta B^T B) (\tilde{y}^k - y^k)\} \geq 0, \quad \forall y \in Y. \end{aligned}$$

Also by the definition of $\tilde{\lambda}^k$, the above equation can be written as:

$$g(y) - g(\tilde{y}^k) + (y - \tilde{y}^k)^T \{-B^T \tilde{\lambda}^k + \tau_k r \beta (\tilde{y}^k - y^k)\} \geq 0, \quad \forall y \in Y. \quad (36)$$

By the definition of $\tilde{\lambda}^k$ in (19), we have:

$$(A\tilde{x} + B\tilde{y}^k - b) - B(\tilde{y}^k - y^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0.$$

The above equation can be written as:

$$\begin{aligned} \tilde{\lambda}^k \in R^m, \quad & \left(\lambda - \tilde{\lambda}^k \right)^T \left\{ A\tilde{x}^k + B\tilde{y}^k - b - B(\tilde{y}^k - y^k) \right. \\ & \left. + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \right\} \geq 0, \quad \forall \lambda \in R^m. \end{aligned} \quad (37)$$

The lemma can be proven by combining equations (35), (36), and (37) along with the notation Q_{k+1} .

Lemma 3.2. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1 and $\{\tilde{w}^k\}$ be defined in (19), then:

$$v^k - v^{k+1} = M(v^k - \tilde{v}^k) \quad (38)$$

where M is defined in (17).

Proof: This follows from (31) and the definitions of $\tilde{\lambda}^k$ and $\hat{\lambda}^k$:

$$\begin{aligned} \lambda^{k+1} &= \lambda^k - \sigma(\lambda^k - \hat{\lambda}^k) \\ &= \lambda^k - \sigma\beta(A\tilde{x}^k + B\hat{y}^k - b) \\ &= \lambda^k - \sigma[\beta(A\tilde{x}^k + By^k - b) - \beta B(y^k - \hat{y}^k)] \\ &= \lambda^k - \sigma(\lambda^k - \tilde{\lambda}^k) + \sigma\beta B(y^k - \hat{y}^k). \end{aligned}$$

This can be seen in combination with (31):

$$\begin{pmatrix} y^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} \sigma I_{n_2} & O \\ -\sigma\beta B & \sigma I_m \end{pmatrix} \begin{pmatrix} y^k - \hat{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

Therefore, equation (38) holds.

It is clear from (38) and (21):

$$\begin{aligned} (v - \tilde{v}^k)^T Q_{k+1} (v^k - \tilde{v}^k) &= (v - \tilde{v}^k)^T H_{k+1} M (v^k - \tilde{v}^k) \\ &= (v - \tilde{v}^k)^T H_{k+1} (v^k - v^{k+1}). \end{aligned}$$

Lemma 3.3. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1 and $\{\tilde{w}^k\}$ be defined in (19), Then, we have:

$$\begin{aligned} (v - \tilde{v}^k)^T H_{k+1} (v^k - v^{k+1}) &= \frac{1}{2} \left(\|v - v^{k+1}\|_{H_{k+1}}^2 - \|v - v^k\|_{H_{k+1}}^2 \right) \\ &+ \frac{1}{2} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2, \quad \forall v \in \Omega. \end{aligned} \quad (39)$$

where the matrix $G_{k+1} = Q_k^T + Q_k - M^T H_{k+1} M$.

Proof: Using the equation:

$$\begin{aligned} (a-b)^T H_{k+1} (c-d) &= \frac{1}{2} \left(\|a-d\|_{H_{k+1}}^2 - \|a-c\|_{H_{k+1}}^2 \right) \\ &\quad + \frac{1}{2} \left(\|c-b\|_{H_{k+1}}^2 - \|d-b\|_{H_{k+1}}^2 \right). \end{aligned} \quad (40)$$

In (40), let $a = v, b = \tilde{v}^k, c = v^k, d = v^{k+1}$. We can get:

$$\begin{aligned} (v - \tilde{v}^k)^T H_{k+1} (v^k - v^{k+1}) &= \frac{1}{2} \left(\|v - v^{k+1}\|_{H_{k+1}}^2 - \|v - v^k\|_{H_{k+1}}^2 \right) \\ &\quad + \frac{1}{2} \left(\|v^k - \tilde{v}^k\|_{H_{k+1}}^2 - \|v^{k+1} - \tilde{v}^k\|_{H_{k+1}}^2 \right). \end{aligned} \quad (41)$$

For the last term in (41), we have:

$$\begin{aligned} &\|v^k - \tilde{v}^k\|_{H_{k+1}}^2 - \|v^{k+1} - \tilde{v}^k\|_{H_{k+1}}^2 \\ &= \|v^k - \tilde{v}^k\|_{H_{k+1}}^2 - \|(v^k - \tilde{v}^k) - (v^k - v^{k+1})\|_{H_{k+1}}^2 \\ &\stackrel{(44)}{=} \|v^k - \tilde{v}^k\|_{H_{k+1}}^2 - \|(v^k - \tilde{v}^k) - M(v^k - \tilde{v}^k)\|_{H_{k+1}}^2 \\ &= 2(v^k - \tilde{v}^k)^T H_{k+1} M (v^k - \tilde{v}^k) - (v^k - \tilde{v}^k)^T M^T H_{k+1} M (v^k - \tilde{v}^k) \\ &= (v^k - \tilde{v}^k)^T (Q_k^T + Q_k - M^T H_{k+1} M) (v^k - \tilde{v}^k). \end{aligned} \quad (42)$$

Combining (41) and (42), (39) is proved.

Now, let us examine the properties of the matrix G_{k+1} . Utilizing (21), we can derive:

$$\begin{aligned} G_{k+1} &= Q_{k+1}^T + Q_{k+1} - M^T H_{k+1} M \\ &= Q_{k+1}^T + Q_{k+1} - M^T Q_{k+1} \\ &= \begin{pmatrix} 2\tau_k r \beta I_{n_2} & -B^T \\ -B & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} \sigma I_{n_2} & -\sigma \beta B^T \\ O & \sigma I_m \end{pmatrix} \begin{pmatrix} \tau_k r \beta I_{n_2} & O \\ -B & \frac{1}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} 2\tau_k r \beta I_{n_2} & -B^T \\ -B & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} \sigma \tau_k r \beta I_{n_2} + \sigma \beta B^T B & -\sigma B^T \\ -\sigma B & \frac{\sigma}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} (2-\sigma) \tau_k r \beta I_{n_2} - \sigma \beta B^T B & (\sigma-1) B^T \\ (\sigma-1) B & \frac{(2-\sigma)}{\beta} I_m \end{pmatrix}. \end{aligned} \quad (43)$$

To simplify the proof, we give some properties of the parameters as follows:

From the iterative format of Algorithm 1: $\frac{1}{1+\eta_k} \tau_k \leq \tau_{k+1} \leq (1+\xi_k) \tau_k$, That is, the sequence in Algorithm 1 satisfies: $\tau_k \in [\tau_{min}, \tau_{max}]$.

Let the $(k+1)$ th step ultimately satisfy the parameters of Step 4 as $\tau_{k+1} = \rho \gamma^{m_k} \tau_k$, where m_k is an integer.

Let $\xi_k := \rho\gamma^{m_k} - 1$, that is $1 + \xi_k = \rho\gamma^{m_k}$.
then:

$$\tau_{k+1} \geq \frac{1 + \xi_k}{1 + \eta_k} \tau_k \geq \dots \geq \frac{\prod_{i=1}^k (1 + \xi_i)}{\prod_{i=1}^k (1 + \eta_i)} \tau_1 \geq \frac{\prod_{i=1}^k (1 + \xi_i)}{\prod_{i=1}^{\infty} (1 + \eta_i)} \tau_0.$$

By parameter setting $\sum_{k=1}^{\infty} \eta_k < +\infty$ know $\prod_{i=1}^{\infty} (1 + \eta_i) < +\infty$.

Making $k \rightarrow +\infty$ gives: $\prod_{i=1}^{\infty} (1 + \xi_i) < +\infty$, that is: $\sum_{k=1}^{\infty} \xi_i < +\infty$.

Theorem 3.1. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1 and $\{\tilde{w}^k\}$ be defined in (19). Then, we have:

$$\begin{aligned} \|v^{k+1} - v^*\|_{H_{k+1}}^2 &\leq (1 + \xi_k) \|v^k - v^*\|_{H_k}^2 - \frac{1}{\sigma^2} \left\{ \beta \|y^k - y^{k+1}\|_{T_{k+1}}^2 \right. \\ &\quad \left. + \frac{(2 - \sigma) - \varepsilon}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right\}. \end{aligned} \quad (44)$$

where $\varepsilon \in (0, 2 - \sigma)$ and $T_{k+1} \succ 0$.

Proof: In the following, we will further investigate the $\|v^k - \tilde{v}^k\|_{G_{k+1}}^2$ term and show how it can be bounded.

Due to $G_{k+1} = \begin{pmatrix} (2 - \sigma) \tau_k r \beta I_{n_2} - \sigma \beta B^T B & (\sigma - 1) B^T \\ (\sigma - 1) B & \frac{(2 - \sigma)}{\beta} I_m \end{pmatrix}$ and $v = \begin{pmatrix} y \\ \lambda \end{pmatrix}$, We have:

$$\begin{aligned} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2 &= (2 - \sigma) \tau_k r \beta \|y^k - \tilde{y}^k\|^2 - \sigma \beta \|B(y^k - \tilde{y}^k)\|^2 \\ &\quad + \frac{2 - \sigma}{\beta} \|\lambda^k - \tilde{\lambda}^k\|^2 + 2(\sigma - 1) (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \\ &= (2 - \sigma) \tau_k r \beta \|y^k - \tilde{y}^k\|^2 + (2 - \sigma) \beta \|A\tilde{x}^k + B\tilde{y}^k - b\|^2 \\ &\quad + 2\beta (A\tilde{x}^k + B\tilde{y}^k - b)^T B(y^k - \tilde{y}^k) \end{aligned} \quad (45)$$

Also because $\hat{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k - b)$ and $y^k - \tilde{y}^k = \frac{1}{\sigma}(y^k - y^{k+1})$.
Therefore, we have:

$$\begin{aligned} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2 &= \frac{(2 - \sigma)}{\sigma^2} \tau_k r \beta \|y^k - y^{k+1}\|^2 + \frac{(2 - \sigma)}{\beta} \|\lambda^k - \hat{\lambda}^k\|^2 \\ &\quad + \frac{2}{\sigma} (\lambda^k - \hat{\lambda}^k)^T B(y^k - y^{k+1}). \end{aligned} \quad (46)$$

It is clear from (31):

$$\lambda^k - \hat{\lambda}^k = \frac{1}{\sigma} (\lambda^k - \lambda^{k+1}).$$

Substituting the above equation into (46) gives:

$$\begin{aligned} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2 &= \frac{(2 - \sigma)}{\sigma^2} \tau_k r \beta \|y^k - y^{k+1}\|^2 + \frac{(2 - \sigma)}{\sigma^2 \beta} \|\lambda^k - \lambda^{k+1}\|^2 \\ &\quad + \frac{2}{\sigma^2} (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}). \end{aligned} \quad (47)$$

In the following, we estimate $(\lambda^k - \lambda^{k+1})^T B (y^k - y^{k+1})$.
From the *Cauchy – Schwarz* inequality: for $\forall \delta > 0$, there is :

$$(\lambda^k - \lambda^{k+1})^T (By^k - By^{k+1}) \geq -\frac{\delta}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 - \frac{1}{4\delta} \beta \|B (y^k - y^{k+1})\|^2. \quad (48)$$

Substituting (48) into (47) shows that:

$$\begin{aligned} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2 &\geq \frac{1}{\sigma^2} \left[(2 - \sigma) \tau_k r \beta \|y^k - y^{k+1}\|^2 - \frac{1}{2\delta} \beta \|B (y^k - y^{k+1})\|^2 \right. \\ &\quad \left. + \frac{(2 - \sigma) - 2\delta}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right]. \end{aligned} \quad (49)$$

Owing to:

$$(w_1 - w_2)^T (F(w_1) - F(w_2)) \geq 0, \quad \forall w_1, w_2 \in \Omega. \quad (50)$$

It follows from (50) and (34):

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(w) \geq (v - \tilde{v}^k)^T Q_{k+1} (v^k - \tilde{v}^k), \quad \forall w \in \Omega.$$

Let the above equation $w = w^*$ and combine with (15) to show that:

$$(\tilde{v}^k - v^*)^T Q_{k+1} (v^k - \tilde{v}^k) \geq 0, \quad \forall w \in \Omega. \quad (51)$$

It is clear from (21) and (38):

$$(\tilde{v}^k - v^*)^T Q_{k+1} (v^k - \tilde{v}^k) = (\tilde{v}^k - v^*)^T H_{k+1} (v^k - v^{k+1}). \quad (52)$$

It is clear from (39):

$$\begin{aligned} (\tilde{v}^k - v^*)^T H_{k+1} (v^k - v^{k+1}) &= \frac{1}{2} \left(\|v^k - v^*\|_{H_{k+1}}^2 - \|v^{k+1} - v^*\|_{H_{k+1}}^2 \right) \\ &\quad - \frac{1}{2} \|v^k - \tilde{v}^k\|_{G_{k+1}}^2. \end{aligned} \quad (53)$$

It follows from (51), (52) and (53):

$$\|v^{k+1} - v^*\|_{H_{k+1}}^2 \leq \|v^k - v^*\|_{H_{k+1}}^2 - \|v^k - \tilde{v}^k\|_{G_{k+1}}^2. \quad (54)$$

Substituting (49) into (54) gives:

$$\begin{aligned}
\|v^{k+1} - v^*\|_{H_{k+1}}^2 &\leq \|v^k - v^*\|_{H_{k+1}}^2 - \frac{1}{\sigma^2} \left\{ \beta \|y^k - y^{k+1}\|_{T_{k+1}}^2 \right. \\
&\quad \left. + \frac{(2-\sigma) - 2\delta}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right\} \\
&\leq \|v^k - v^*\|_{H_k}^2 + (\tau_k - \tau_{k-1}) \frac{r\beta}{\sigma} \|y^k - y^*\|^2 \\
&\quad - \frac{1}{\sigma^2} \left\{ \beta \|y^k - y^{k+1}\|_{T_{k+1}}^2 + \frac{(2-\sigma) - 2\delta}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right\} \\
&\leq (1 + \xi_k) \|v^k - v^*\|_{H_k}^2 - \frac{1}{\sigma^2} \left\{ \beta \|y^k - y^{k+1}\|_{T_{k+1}}^2 \right. \\
&\quad \left. + \frac{(2-\sigma) - 2\delta}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right\}
\end{aligned} \tag{55}$$

where $T_{k+1} = (2-\sigma)\tau_k r I_{n_2} - \frac{1}{2\delta} B^T B$.

For $\lim_{k \rightarrow \infty} (v^k - v^{k+1}) = 0$ to hold, it is sufficient that: $\delta = \frac{\varepsilon}{2}$ and $T_{k+1} \succ 0$.

$T_{k+1} \succ 0$ is equivalent to $(2-\sigma)r_k \|y^k - y^{k+1}\|^2 > \frac{1}{\varepsilon} \|B(y^k - y^{k+1})\|^2$.

So (55) can be written as:

$$\begin{aligned}
\|v^{k+1} - v^*\|_{H_{k+1}}^2 &\leq (1 + \eta_k) \|v^k - v^*\|_{H_k}^2 - \frac{1}{\sigma^2} \left\{ \beta \|y^k - y^{k+1}\|_{D_{k+1}}^2 \right. \\
&\quad \left. + \frac{(2-\sigma) - \varepsilon}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 \right\}.
\end{aligned}$$

Therefore, Theorem 3.1 is proved.

Theorem 3.2. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1. Taking any point w^* in Ω^* , we have:

(a). $\lim_{k \rightarrow \infty} \|v^{k+1} - v^*\|_{H_{k+1}}^2$ exists and $\lim_{k \rightarrow \infty} \|v^{k+1} - v^*\|_{H_{k+1}}^2 < +\infty$.

(b). $\lim_{k \rightarrow \infty} \|y^{k+1} - y^k\| = 0$ and $\lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = 0$.

Proof: Let

$$\begin{aligned}
a^k &= \|v^{k+1} - v^*\|_{H_{k+1}}^2, b^k = \xi_k, c^k = 0, \\
d^k &= \beta \|y^{k+1} - y^k\|_{D_{k+1}}^2 + \frac{1-2\varepsilon}{\beta} \|\lambda^{k+1} - \lambda^k\|^2.
\end{aligned}$$

By Lemma 2.2, (a) and (b) hold.

Theorem 3.3. Let the sequence $\{w^k\}$ be the iterative sequence generated by Algorithm 1, then $\{w^k\}$ converges to a point $w^\infty \in \Omega^*$.

Proof: It follows from Theorem 3.2: $\lim_{k \rightarrow \infty} \|v^{k+1} - v^k\| = 0$.

It follows from combining (38) and the non-singularity of M : $\lim_{k \rightarrow \infty} \|v^k - \tilde{v}^k\| = 0$.

Since the sequence $\left\{ \|v^{k+1} - v^*\|_{H_{k+1}}^2 \right\}$ is a bounded sequence,

then for any fixed $v^* \in \Omega^*$, there is a $\|v^{k+1} - v^*\|$ bounded.

That is, the sequence $\{v^k\}$ is bounded.
because of

$$\|\tilde{v}^k - v^*\| \leq \|v^k - \tilde{v}^k\| + \|v^k - v^*\|.$$

It is known that $\|\tilde{v}^k - v^*\|$ is bounded.

Clearly the sequence $\{\tilde{v}^k\}$ is also bounded and there must exist a convergence point v^∞ ,

such that a subsequence $\{\tilde{v}^{k_j}\}$ of the existence sequence $\{\tilde{v}^k\}$ converges to v^∞ .

From $\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b)$ in (19), we have:

$$A\tilde{x}^{k_j} = \frac{1}{\beta}(\lambda^{k_j} - \tilde{\lambda}^{k_j}) - (By^{k_j} - b).$$

Since Matrix A is a column full rank matrix, it follows that the sequence $\{\tilde{x}^{k_j}\}$ converges.

Set $\lim_{k_j \rightarrow \infty} \tilde{x}^{k_j} = x^\infty$, then there exists a subsequence $\{\tilde{w}^{k_j}\}$ converging to w^∞ .

Let $k = k_j$ in (34), we know that $\tilde{w}^{k_j} \in \Omega$,

$$\theta(w) - \theta(\tilde{w}^{k_j}) + (w - \tilde{w}^{k_j})^T F(\tilde{w}^{k_j}) \geq (w - \tilde{w}^{k_j})^T Q(w^{k_j} - \tilde{w}^{k_j}), \quad \forall w \in \Omega.$$

Let the above equation $k \rightarrow \infty$, it is clear that:

$$w^\infty \in \Omega, \theta(w) - \theta(w^\infty) + (w - w^\infty)^T F(w^\infty) \geq 0, \forall w \in \Omega. \quad (56)$$

(56) shows that w^∞ in Ω^* is a solution of the variational inequality $VI(\Omega, F, \theta)$, so the sequence $\{w^k\}$ converges to w^∞ .

4 Numerical experiments

The numerical performance of Algorithm 1 is presented in this paper through solving the Lasso problem and comparing Algorithm 1 with optimal linearized ADMM(OLADMM)[22]. All simulation experiments were conducted on a laptop with 4GB of RAM memory using Matlab R2016a.

The LASSO problem[12] in statistics is as follows:

$$\min_{x,y} \frac{1}{2} \|x - b\|_2^2 + \iota \|y\|_1 \quad s.t. \quad x = Ay \quad (57)$$

where $A \in R^{m \times n}, b \in R^m, x \in R^n, y \in R^n$.

Then applying Algorithm 1 to (57), we obtain:

The x -subproblem is an unconstrained convex quadratic minimization problem, and its unique solution can be expressed as follows:

$$x^{k+1} = \frac{1}{1 + \beta} (b + \lambda^k + \beta Ay^k)$$

The y -subproblem is a $l_1 + l_2$ minimization problem that can be solved using the soft-threshold operator:

$$\hat{y}^k = \mathit{shrink} \left\{ y^k - \frac{1}{\tau_k \delta_k \beta} A^T [\lambda^k - \beta (x^{k+1} - Ay^k)], \frac{\iota}{\tau_k \delta_k \beta} \right\}.$$

$$y^{k+1} = y^k - \sigma (y^k - \hat{y}^k).$$

The solution to the λ -subproblem is:

$$\hat{\lambda}^k = \lambda^k - \beta (x^{k+1} - A\hat{y}^k).$$

$$\lambda^{k+1} = \lambda^k - \sigma (\lambda^k - \hat{\lambda}^k).$$

In this paper, numerical experiments are conducted to compare the performance of the Algorithm 1 with OLADMM .

Set the parameters of each algorithm as follows:

Algorithm 1: $r = \|A^T A\|$, $\iota = 0.1 \|A^T b\|_\infty$, $\tau_0 = 0.75$, $\tau_{min} = 0.01$, $\gamma = 1.2$, $\sigma = 0.9$, $\beta = 1$, $\rho = 3$, $p^0 = 100$, $d^0 = 100$, $\varepsilon' = \frac{1}{\varepsilon} = \frac{1}{2-\sigma} + 0.1$, $\eta_k = 0.25 \min \left\{ 1, \frac{1}{(\max\{1, k-l\})^2} \right\}$, $s_k = 2 \min \left\{ 1, \frac{1}{(\max\{1, k-l\})^2} \right\}$.

where l = the dimension of λ .

OLADMM: $r = \|A^T A\|$, $\tau = 0.75$, $\beta = 1$, $\iota = 0.1 \|A^T b\|_\infty$.

The stopping guidelines are:

$$\|p^{k+1}\| = \|x^{k+1} - Ay^{k+1}\| < \varepsilon^{pri}, \quad \text{and} \quad \|d^{k+1}\| = \|\beta A (y^{k+1} - y^k)\| < \varepsilon^{dual}.$$

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} set to be 10^{-4} and 10^{-2} .

For a matrix A of given dimension $m \times n$, we generate the data randomly as follow: $p = \frac{1}{n}$, $x^0 = \mathit{sprandn}(n, 1, p)$, $A = \mathit{randn}(m, n)$, $b = A * x^0 + \mathit{sqr}(0.001) * \mathit{randn}(m, 1)$. The initial point is $(y^0, \lambda^0) = (0, 0)$.

In order to observe the effect on the experiment caused by different values of σ , the choice of the:

$$\sigma = \{0.1, 0.2, 0.3, \dots, 1.7, 1.8, 1.9\}$$

to carry out the experiment, the results of which show that when $\sigma \in [0.6, 1.4]$, the results are satisfactory. In this paper, we take $\sigma = 0.9$.

The experimental results of the two algorithms are shown below:
Table 1. Comparison between Algorithm 1 and OLADMM for (57).

$n \times n$ matrix		OLADMM				Algorithm 1			
m	n	Iter.	CPU(s)	$\ p^k\ $	$\ q^k\ $	Iter.	CPU(s)	$\ p^k\ $	$\ q^k\ $
1000	1500	16	3.23	0.0700	0.0701	11	2.38	0.0414	0.0446
1500	1500	13	2.81	0.0650	0.0660	10	2.53	0.0350	0.0500
1500	3000	17	7.03	0.0658	0.0659	10	6.18	0.0448	0.0599
2000	3000	14	7.16	0.0759	0.0764	10	6.57	0.0243	0.0399
3000	3000	13	7.80	0.0572	0.0582	9	7.40	0.0377	0.0695
3000	5000	15	24.96	0.0685	0.0688	10	23.44	0.0312	0.0485
4000	5000	13	25.01	0.0664	0.0674	10	24.69	0.0365	0.0549
5000	5000	12	26.34	0.0614	0.0636	9	25.73	0.0435	0.0739

Table 1 presents the number of iterations and total computation time needed for OLADMM and Algorithm 1. It is evident that Algorithm 1 consistently outperforms OLADMM as it demands fewer iteration steps and less time to meet the termination condition. Consequently, Algorithm 1 demonstrates superior efficiency in contrast to OLADMM. Furthermore, Algorithm 1 significantly surpasses OLADMM, providing robust backing for our convergence analysis.

5 Conclusion

In this study, we introduce an adaptive linearized alternating direction multiplier method with a relaxation step that incorporates both a relaxation step and an adaptive technique. Our approach adopts a unified framework of variational inequalities and establishes the global convergence of this adaptive linearized alternating direction multiplier method with a relaxation step through theoretical analysis. Through the resolution of the Lasso problem, we demonstrate the algorithm’s outstanding numerical performance. These findings will propel advancements in optimization algorithms and offer more efficient tools and methodologies for addressing practical challenges.

Conflict of Interest:The authors declare that they have no conflict of interest.

References

- [1] Tao, M., Yang, J., He, B.: Alternating direction algorithms for total variation deconvolution in image reconstruction. TR0918, Department of Mathematics, Nanjing University (2009)
- [2] Ng, M.K., Weiss, P., Yuan, X.: Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM journal on Scientific Computing* **32**(5), 2710–2736 (2010)
- [3] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., *et al.*: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* **3**(1), 1–122 (2011)

- [4] Combettes, P.L., Pesquet, J.-C.: A douglas–rachford splitting approach to non-smooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing* **1**(4), 564–574 (2007)
- [5] Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* **2**(1), 17–40 (1976)
- [6] Glowinski, R., Marrocco, A.: Analyse numerique du champ magnetique d’un alternateur par elements finis et sur-relaxation ponctuelle non lineaire. *Computer Methods in Applied Mechanics & Engineering* **3**(1), 55–85 (1974)
- [7] He, B.: My 20 years research on alternating directions method of multipliers. *Oper. Res. Trans* **22**(1), 1–31 (2018)
- [8] Eckstein, J., Yao, W.: Approximate admm algorithms derived from lagrangian splitting. *Computational Optimization and Applications* **68**, 363–405 (2017)
- [9] He, B., Liao, L.-Z., Han, D., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming* **92**, 103–118 (2002)
- [10] Ng, M.K., Wang, F., Yuan, X.: Inexact alternating direction methods for image recovery. *SIAM Journal on Scientific Computing* **33**(4), 1643–1668 (2011)
- [11] Chan, R.H., Tao, M., Yuan, X.: Linearized alternating direction method of multipliers for constrained linear least-squares problem. *East Asian Journal on Applied Mathematics* **2**(4), 326–341 (2012)
- [12] Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **58**(1), 267–288 (1996)
- [13] Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* **52**(3), 471–501 (2010)
- [14] Tao, M., Yuan, X.: Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization* **21**(1), 57–81 (2011)
- [15] Fang, E.X., He, B., Liu, H., Yuan, X.: Generalized alternating direction method of multipliers: new theoretical insights and applications. *Mathematical programming computation* **7**(2), 149–187 (2015)
- [16] Woo, H., Yun, S.: Proximal linearized alternating direction method for multiplicative denoising. *SIAM Journal on Scientific Computing* **35**(2), 336–358 (2013)

- [17] He, B., Yuan, X.: Linearized alternating direction method of multipliers with gaussianback substitution for separable convex programming. *Numerical Algebra, Control and Optimization* **3**(2), 247–260 (2013)
- [18] Ouyang, Y., Chen, Y., Lan, G., Pasiliao Jr, E.: An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences* **8**(1), 644–681 (2015)
- [19] Chan, R.H., Tao, M., Yuan, X.: Linearized alternating direction method of multipliers for constrained linear least-squares problem. *East Asian Journal on Applied Mathematics* **2**(4), 326–341 (2012)
- [20] Wang, X., Yuan, X.: The linearized alternating direction method of multipliers for dantzig selector. *SIAM Journal on Scientific Computing* **34**(5), 2792–2811 (2012)
- [21] He, B., Ma, F., Yuan, X.: Linearized alternating direction method of multipliers via positive-indefinite proximal regularization for convex programming. Available on (2016) <https://doi.org/https://optimization-online.org>
- [22] He, B., Ma, F., Yuan, X.: Optimally linearizing the alternating direction method of multipliers for convex programming. *Computational Optimization and Applications* **75**(2), 361–388 (2020)
- [23] He, B.: Ppa-like contraction methods for convex optimization: a framework using variational inequality approach. *Journal of the Operations Research Society of China* **3**, 391–420 (2015)
- [24] Tao, M., Yuan, X.: On the optimal linear convergence rate of a generalized proximal point algorithm. *Journal of Scientific Computing* **74**, 826–850 (2018)
- [25] Gu, G., Yang, J.: On the optimal linear convergence factor of the relaxed proximal point algorithm for monotone inclusion problems. arXiv preprint arXiv:1905.04537 (2019)
- [26] Robbins H, S.D.: A convergence theorem for non negative almost supermartingales and some applications. *Optimizing Methods in Statistics*, 233–257 (1971)

