A PROXIMAL ALTERNATING DIRECTION METHOD FOR MULTI-BLOCK COUPLED CONVEX OPTIMIZATION

FOXIANG LIU
Institute of Electromagnetics and Acoustics
Department of Electronic Science
Xiamen University, Xiamen, 361005, China
and
School of Mathematical Sciences, Nanjing Normal University
Jiangsu Key Laboratory for NSLCS
Nanjing Normal University, Nanjing 210023, China

LINGLING XU*, YUEHONG SUN AND DEREN HAN
School of Mathematical Sciences, Nanjing Normal University
Jiangsu Key Laboratory for NSLCS
Nanjing Normal University, Nanjing 210023, China

(Communicated by Kok Lay Teo)

ABSTRACT. In this paper, we propose a proximal alternating direction method (PADM) for solving the convex optimization problems with linear constraints whose objective function is the sum of multi-block separable functions and a coupled quadratic function. The algorithm generates the iterate via a simple correction step, where the descent direction is based on the PADM. We prove the convergence of the generated sequence under some mild assumptions. Finally, some familiar numerical results are reported for the new algorithm.

1. Introduction. In this paper, we consider the following problem:

\[
\min_{x \in R^d} \theta(x) := \sum_{i=1}^{N} \theta_i(x_i) + \frac{1}{2} x^T H x + g^T x
\]

subject to 
\[\sum_{i=1}^{N} A_i x_i = b, \quad x_i \in \mathcal{X}_i, \quad i = 1, \ldots, N,\]  

where \( \theta_i : R^{d_i} \mapsto (-\infty, +\infty] \) are closed convex (not necessarily smooth) functions; \( x_i \in R^{d_i} \), \( x = (x_1, x_2, \ldots, x_N) \in R^d \); \( H \in R^{d \times d} \) is a symmetric positive semidefinite matrix; \( g \in R^d \); \( A_i \in R^{m \times d_i} \), and \( b \in R^m \).

2010 Mathematics Subject Classification. Primary: 90C25, 65K25; Secondary: 58E35.
Key words and phrases. Convex optimization, correction step, multi-block, proximal alternating direction method, coupled quadratic function.

The second author is supported by the National Natural Science Foundation of China [Grant No. 11401314].

* Corresponding author.
This kind of problem has many applications in signal and imaging processing, machine learning, management, statistics and engineering computation, traffic management, cloud services and biological data. For example, many cloud traffic management problems [14] can be transformed into the following form:

\[
\max \sum_{i=1}^{N} \psi_i(x_{i1}, \ldots, x_{in}) - \sum_{j=1}^{n} \phi_j(y_j)
\]

s.t. \[ \sum_{i=1}^{N} x_{ij} = y_j, \ j = 1, 2, \ldots, n, \]

\[ x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in \mathcal{X}_i \subseteq \mathbb{R}^n, \ y_j \in \mathcal{Y} \subseteq \mathbb{R}. \]

Here each \( \psi_i(x_{i1}, \ldots, x_{in}) \) represents the "level of satisfaction" of user \( i \) when she receives an amount \( x_{ij} \) of resources from facility \( j \), and each \( \psi_i \) is a concave function. Generally, \( \psi_i \) is a quadratic function. \( \phi_j(y_j) \) represents the operational expense or congestion cost when facility \( j \) allocates an amount \( y_j \) of resources to all the users. All the functions \( \phi_j (j = 1, \ldots, n) \) are convex. To learn more applications, please refer to [1, 8, 27, 33, 34], etc., for details.

The augmented Lagrangian function of problem (1) is

\[
L_\beta(x_1, x_2, \ldots, x_N; \lambda) := \sum_{i=1}^{N} \theta_i(x_i) + \frac{1}{2} x^T H x + g^T x - \lambda^T (\sum_{i=1}^{N} A_i x_i - b) + \frac{\beta}{2} \| \sum_{i=1}^{N} A_i x_i - b \|^2,
\]

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

When the coupled objective function vanishes (\( H = 0 \) and \( g = 0 \)), the problem (1) is reduced to a separable convex programming with linear constraints, which can be solved successfully by many methods. Among these methods, alternating direction method of multiplier (ADMM) is very efficient for solving large scale problems, which solves the problems with the following iterative scheme:

\[
\begin{align*}
\begin{cases}
  x_{k+1}^1 := \arg \min_{x_1} L_\beta(x_1, x_2^k, \ldots, x_N^k, \lambda_k), \\
  x_{k+1}^2 := \arg \min_{x_2} L_\beta(x_1^{k+1}, x_2, \ldots, x_N^k, \lambda_k), \\
  \vdots \\
  x_{k+1}^N := \arg \min_{x_N} L_\beta(x_1^{k+1}, x_2^{k+1}, \ldots, x_N, \lambda_k),
\end{cases}
\end{align*}
\]

(3)

and updates the multiplier via

\[
\lambda_{k+1} := \lambda_k - \tau \beta (\sum_{i=1}^{N} A_i x_i^{k+1} - b).
\]

(4)

where \( \tau > 0 \) is a positive constant. The classic 2-block ADMM algorithm was originally introduced in early 1970s [16, 18], and the convergence has been studied extensively under various conditions and different models. When \( \theta_1 \) is strong convex, \( A_1 \) is identity mapping and \( A_2 \) is injective, [16] first proved the convergence of the classic 2-block ADMM for any \( \tau \in (0, 2) \) if \( \theta_2 \) is linear; When \( \theta_2 \) is a general nonlinear convex function, the convergence was introduced by [15] for any \( \tau \in (0, (1 + \sqrt{5})/2) \). For the case of \( N \geq 3 \), [5] showed that multi-block ADMM is
not necessarily convergent by a counterexample. Many researchers put forward
other kinds of conditions or used proximal terms and correction steps to ensure
the convergence for solving a certain class of problem. For example, the authors of
[24] proposed a Gaussian substitution method, which is a prediction-correction type
method whose predictor was generated by the ADMM procedure and the correction
was completed by a Gaussian back substitution procedure. In [26], a Jacobian-like
method was presented. Each variant was generated by fixing other variants to the
last iterates at the prediction step. Hence a parallel computation can be taken at
each iteration, which is particularly suitable for large-scale problems. For details of
prediction-correction method, please refer to [21, 22, 24, 26] etc. and the references
therein.

If the penalty parameter $\beta$ is limited in a specific range and at least N-2 functions
are strongly convex, the convergence of multi-block ADMM was proved by [4, 20, 7].
If certain error bound condition is satisfied, sufficient small dual stepsize guaranteed
linearly convergent of multi-block ADMM [29]. The convergence rate of ADMM
was also done by [9, 10, 32, 29]. Recently it has become widely popular in big data
related problems arising in machine learning, computer vision, signal processing,
and so on; see [3, 11, 12, 19] and the references therein.

In this paper, we consider how to solve (1) while $H \neq 0$, i.e. the objective
function is not separable. Only few papers focused on the model in the literature
[6, 8, 17, 27, 37], etc. In [6], the authors proposed a randomly permuted ADMM
(RPADMM) to solve the nonseparable multi-block convex optimization, and proved
its expected convergence while applying to solve a class of quadratic programming
problems. In [17], the authors solved the problem

$$\min_x \theta(x) := \sum_{i=1}^{N} \theta_i(x_i) + f(x_1, x_2, \cdots, x_N) \tag{5}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b, \quad x_i \in \mathcal{X}_i, \quad i = 1, \cdots, N,$$

by ADMM, where $\theta_i : R^{d_i} \mapsto (-\infty, +\infty]$ are all strongly convex and $f(\cdot)$ is convex
function with Lipschitz continuous gradient. An upper bound minimization method
was analyzed in [27]. During each iteration, an approximate augmented Lagrangian
of the original problem was presented to avoid using the couple function directly.
The algorithm updated the dual variable by a gradient ascent step followed by
a block coordinate descent (BCD) step for a certain approximate version of the
augmented Lagrangian. The convergence analysis depended on strict condition
of the couple function and the chosen approximate function. In fact, choosing a
proper approximate function for the algorithm is not an easy work due to strict
conditions. Similar idea appeared in [8], which introduced majorization techniques
and considered the 2-block case for general couple function. If the couple function
is merely quadratic and separably smooth, majorized ADMM [8] is exactly same as
the one proposed by [27] under a proper choice of the majorization function.

As said in [28], “when the objective function is not separable across the variables,
the convergence of the ADMM is still open, even in the case where $N = 2$ and each
$\theta_i$ is convex.” We want to know when ADMM can converge while solving such
nonseparable problem, and how to implement it efficiently. Based on the work of
[37], we use a correction step to obtain a descent direction for multi-block problems,
and then focus on analyzing the 3-block problems. Our algorithm need’t choose proper majorization function or require strict assumptions.

This paper is organized as follows. In section 2, we give some notations and introduce our algorithm. In section 3, we discuss the convergence of 3-block PADM. Numerical results are reported in section 4. As last, we present some conclusions in section 5.

In this paper, we denote \( \|x\|_2 = \sqrt{x^T x} \) as the Euclidean-norm and \( \|x\|_G = \sqrt{x^T G x} \) as \( G \)-norm for given symmetric positive definite matrix \( G \). Especially, when \( G \) is symmetric positive semidefinite, we denote \( \|x\|_G = \sqrt{x^T G x} \).

2. A proximal alternating direction method. To present the analysis in a compact way, we give some notations first.

\[
H := \begin{pmatrix}
H_{11} & H_{12} & \cdots & H_{1N} \\
H_{21} & H_{22} & \cdots & H_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
H_{N1} & \cdots & \cdots & H_{NN}
\end{pmatrix}, \quad Q := \begin{pmatrix}
Q_1 \\
Q_2 \\
\vdots \\
Q_N
\end{pmatrix}, \quad (6)
\]

\[
M := \begin{pmatrix}
2H_{11} + Q_1 & 2H_{12} & \cdots & 2H_{1N} \\
2H_{21} - \beta A_2^T A_1 & 2H_{22} + Q_2 & \cdots & 2H_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
H_{N1} - \beta A_N^T A_1 & H_{N2} - \beta A_N^T A_2 & \cdots & 2H_{N-1N} + Q_N
\end{pmatrix}, \quad (7)
\]

\[
S_0 := \begin{pmatrix}
H_{11} + Q_1 & -\beta A_1^T A_2 & \cdots & -\beta A_1^T A_N \\
H_{21} & H_{22} + Q_2 & \cdots & -\beta A_2^T A_N \\
\vdots & \vdots & \ddots & \vdots \\
H_{N1} & H_{N2} & \cdots & H_{NN} + Q_N \\
0 & 0 & \cdots & 0
\end{pmatrix}, \quad S := \frac{S_0 + S_0^T}{2}, \quad (8)
\]

\[
\omega := \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_N \\
\lambda
\end{pmatrix}, \quad g := \begin{pmatrix}
g_1 \\
g_2 \\
\vdots \\
g_N
\end{pmatrix}, \quad P := \begin{pmatrix}
H \\
\frac{1}{\beta} I
\end{pmatrix}, \quad (9)
\]

where \( \beta > 0 \) is a penalty parameter, \( H \) is a symmetric positive semidefinite matrix. We can choose proper matrices \( Q_1, \cdots, Q_N \), which make \( Q \) and \( S \) to be symmetric positive definite.

Xu and Han [37] proposed a proximal alternating direction method (PADM) to solve weakly coupled variational inequalities. In this paper, we extend it to multi-block case and apply it to solve (1). Now we present the \( N \)-block proximal alternating direction method(PADM) as follows:

**Algorithm 1.** A proximal alternating direction method (denoted by PADM).
Choosing the constant \( \beta > 0 \) and matrices \( Q_1, Q_2, \cdots, Q_N \) to be symmetric positive definite. For given iterate scheme \( \omega_k := (x_1^k, x_2^k, \cdots, x_N^k, \lambda^k) \), the new iterate \( \omega_{k+1} := (x_1^{k+1}, x_2^{k+1}, \cdots, x_N^{k+1}, \lambda^{k+1}) \) is generated as follows.
Step 1. (prediction step) Obtain auxiliary iterate scheme \(\tilde{\omega}_k := (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_N^k, \tilde{\lambda}^k)\) by the following procedure

\[
\begin{align*}
\hat{x}_i^k := \arg\min_{x_i} \left\{ \theta(x_1) - (\lambda^k)^T A_1 x_1 + \frac{\beta}{2} \| A_1 x_1 + \sum_{i=2}^{N} A_i x_i^k - b \|^2 \\
+ \frac{1}{2} \| x_1 - x_1^k \|_Q^2 \mid x_1 \in \mathcal{X}_1 \right\}, \\
\tilde{x}_i^k := \arg\min_{x_i} \left\{ \theta(x_i) - (\lambda^k)^T A_i x_i + \frac{\beta}{2} \| \sum_{j=1}^{i-1} A_j \tilde{x}_j^k + A_i x_i + \sum_{j=i+1}^{N} A_j \tilde{x}_j^k - b \|^2 \\
+ \frac{1}{2} \| x_i - \tilde{x}_i^k \|_Q^2 \mid x_i \in \mathcal{X}_i \right\}, i = 2, 3, \ldots, N, \\
\tilde{\lambda}^k := \lambda^k - \beta \sum_{i=1}^{N} A_i \tilde{x}_i^k - b.
\end{align*}
\]

Step 2. (correction step) Correct \(\tilde{\omega}_k\) and generate the new iterate \(\omega_{k+1}\) via:

\[
\omega_{k+1} := \omega_k - \alpha_k M^T(\omega_k - \tilde{\omega}_k),
\]
where

\[
\alpha_k := \frac{\phi(\omega_k, \tilde{\omega}_k)}{\| M^T(\omega_k - \tilde{\omega}_k) \|_2^2},
\]

\(M\) is defined in (7) and

\[
\phi(\omega_k, \tilde{\omega}_k) := \| \omega_k - \tilde{\omega}_k \|_S^2.
\]

Remark 2.1. It is obvious that \(\sum_{i=1}^{N} A_i \tilde{x}_i^k = b\) from (10) when \(\lambda_k = \tilde{\lambda}_k\). That is to say, \((\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_N^k)\) is a feasible solution of (1). According to (10)-(11), we have that \((\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_N^k)\) is a solution of (1) immediately providing that \(\omega_k = \tilde{\omega}_k\). Hence it is reasonable to use \(\| \omega_k - \omega_k \|_S < \varepsilon\) as stopping criterion.

Remark 2.2. In fact, it is easy to choose the matrices \(Q\) and \(S\) such that they are symmetric positive definite. Hence \(\phi(\omega_k, \tilde{\omega}_k)\) can be regarded as the type of \(S\)-norm.

Remark 2.3. In new approach, we achieve the next iterate via using the descent direction \(-M^T(\omega_k - \tilde{\omega}_k)\), we call it correction step. For more details about correction step, please refer [22] and [39].

3. Convergence of 3-block PADM. For convenient expression, we specify \(N = 3\) and analyze the convergence of 3-block PADM for the problem (1). The results in this section can be generalized directly to the case when \(N > 3\). We denote

\[
H := \begin{pmatrix}
H_{11} & H_{12} & H_{13} \\
H_{21} & H_{22} & H_{23} \\
H_{31} & H_{32} & H_{33}
\end{pmatrix}, \quad g := \begin{pmatrix}
g_1 \\
g_2 \\
g_3
\end{pmatrix}.
\]

As a result, the problem (1) for \(N = 3\) can be written as

\[
\min_{x \in \mathbb{R}^3} \theta(x) := \theta_1(x_1) + \theta_2(x_2) + \theta_3(x_3) + \frac{1}{2} x^T H x + g^T x
\]

s.t. \(x = (x_1, x_2, x_3) \in \Omega\), (15)
where
\[ \Omega = \{(x_1, x_2, x_3)|A_1x_1 + A_2x_2 + A_3x_3 = b, \ x_i \in \mathcal{X}_i, \ i = 1, 2, 3\}. \]

**Assumption 3.1.**

(1) \( H \) is a symmetric positive semidefinite matrix.

(2) The solution set \( W^* \) of the problem (15) is nonempty.

(3) Each \( \mathcal{X}_i \) is a closed convex set, \( i = 1, 2, 3 \).

The following lemmas present some contract properties of the 3-block PADM, which play an important role in the subsequent analysis.

**Lemma 3.1.** Suppose that Assumption 3.1 holds. Let \( \omega_k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \bar{\lambda}_k) \) be generated by (10). Then for any \( \omega^* = (x_1^*, x_2^*, x_3^*, \lambda^*) \in W^* \), we have
\[
(\omega_k - \omega^*)^T P(\omega^* - \omega_k) + (\omega_k - \omega^*)^T S_0(\omega_k - \omega_k) \geq \frac{1}{\beta}(\bar{\lambda}_k - \lambda^*)^T (\lambda^* - \lambda_k). \quad (16)
\]
where \( P \) and \( S_0 \) are defined in (8) and (9), respectively.

**Proof.** According to the optimality conditions of (10), and \( x_i^* \in \mathcal{X}_i, i = 1, 2, 3 \), we obtain that
\[
\begin{pmatrix}
    x_1^1 - \hat{x}_1^1 \\
    x_2^1 - \hat{x}_2^1 \\
    x_3^1 - \hat{x}_3^1 \\
    \lambda^* - \bar{\lambda}_k
\end{pmatrix}^T
\begin{pmatrix}
    \nabla x_1^1 L_\beta(\tilde{x}_1^k, x_2^k, x_3^k, \lambda_k) + Q_1(\tilde{x}_1^k - x_1^k) \\
    \nabla x_2^1 L_\beta(\tilde{x}_1^k, \tilde{x}_2^k, x_3^k, \lambda_k) + Q_2(\tilde{x}_2^k - x_2^k) \\
    \nabla x_3^1 L_\beta(\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \lambda_k) + Q_3(\tilde{x}_3^k - x_3^k) \\
    \nabla x_1^1 L_\beta(\tilde{x}_1^k, x_2^k, x_3^k, \lambda_k) + Q_1(\tilde{x}_1^k - x_1^k)
\end{pmatrix} \geq 0. \quad (17)
\]
On the other hand, since \( (x_1^*, x_2^*, x_3^*, \lambda^*) \in W^*, (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \), we have
\[
\begin{pmatrix}
    \hat{x}_1^1 - x_1^* \\
    \hat{x}_2^1 - x_2^* \\
    \hat{x}_3^1 - x_3^* \\
    \bar{\lambda}_k - \lambda^*
\end{pmatrix}^T
\begin{pmatrix}
    \partial \theta_1(x_1^*) + H_{11}x_1^* + H_{12}x_2^* + H_{13}x_3^* + g_1 - A_1^T \lambda^* \\
    \partial \theta_2(x_2^*) + H_{21}x_1^* + H_{22}x_2^* + H_{23}x_3^* + g_2 - A_2^T \lambda^* \\
    \partial \theta_3(x_3^*) + H_{31}x_1^* + H_{32}x_2^* + H_{33}x_3^* + g_3 - A_3^T \lambda^* \\
    A_1\hat{x}_1^1 + A_2\hat{x}_2^1 + A_3\hat{x}_3^1 - b
\end{pmatrix} \geq 0. \quad (18)
\]
Adding (17) and (18), we get
\[
\begin{pmatrix}
    \hat{x}_1^1 - x_1^* \\
    \hat{x}_2^1 - x_2^* \\
    \hat{x}_3^1 - x_3^* \\
    \bar{\lambda}_k - \lambda^*
\end{pmatrix}^T
\begin{pmatrix}
    H_{11}(x_1^1 - \hat{x}_1^1) + H_{12}(x_2^1 - \hat{x}_2^1) + H_{13}(x_3^1 - \hat{x}_3^1) + A_1^T (\lambda_k - \lambda^*) \\
    H_{21}(x_1^1 - \hat{x}_1^1) + H_{22}(x_2^1 - \hat{x}_2^1) + H_{23}(x_3^1 - \hat{x}_3^1) + A_2^T (\lambda_k - \lambda^*) \\
    H_{31}(x_1^1 - \hat{x}_1^1) + H_{32}(x_2^1 - \hat{x}_2^1) + H_{33}(x_3^1 - \hat{x}_3^1) + A_3^T (\lambda_k - \lambda^*) \\
    A_1\hat{x}_1^1 + A_2\hat{x}_2^1 + A_3\hat{x}_3^1 - b
\end{pmatrix} \geq 0. \quad (19)
\]
The last inequality is obtained easily by the convexity of \( \theta_i(x), i = 1, 2, 3 \). From (19), we further get
Combining (8) with (9), and adding the item $(\tilde{A} \lambda - \lambda^*)^T \frac{1}{\beta} (\lambda^* - \tilde{\lambda})$ on both sides of the inequality (20), we get (16) in a compact way.

\begin{flushright}
\text{\ding{51}}
\end{flushright}

**Lemma 3.2.** Suppose that Assumption 3.1 holds. Let $\tilde{\omega}_k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{\lambda}_k)$ be generated by (10). Then for any $\omega^* \in W^*$, we have

$$
(M(\omega_k - \omega^*)) \geq \phi(\omega_k, \tilde{\omega}_k),
$$

where $M$ and $\phi(\omega_k, \tilde{\omega}_k)$ are defined in (7) and (13), respectively.

**Proof.** Since $\tilde{\omega}_k - \omega^* = \tilde{\omega}_k - \omega_k + \omega_k - \omega^*$ and $\tilde{\lambda}_k - \lambda^* = \tilde{\lambda}_k - \lambda_k + \lambda_k - \lambda^*$, from (16), we immediately have

$$
(\omega_k - \tilde{\omega}_k)^T (P + S_0^T)(\omega_k - \omega^*) \geq (\omega_k - \tilde{\omega}_k)^T S_0(\omega_k - \tilde{\omega}_k) + \|\omega_k - \omega^*\|_p^2 + \frac{1}{\beta} (\tilde{\lambda}_k - \lambda^*)^T (\lambda^* - \tilde{\lambda}_k).
$$

(22)

According to (9), we get

$$
\|\omega_k - \omega^*\|_p^2 = x^T H x + \frac{1}{\beta} (\lambda_k - \lambda^*)^T (\lambda_k - \lambda^*).
$$

(23)

Combining (7) with (22), we know

$$
M + \text{diag}(0, \frac{1}{\beta} I) = P + S_0^T.
$$

(24)

Substituting (23) and (24) into (22), we get

$$
(\omega_k - \tilde{\omega}_k)^T M(\omega_k - \omega^*) \geq (\omega_k - \tilde{\omega}_k)^T S(\omega_k - \tilde{\omega}_k)
$$

(25)
which completes the proof. 

**Theorem 3.3.** Suppose that Assumption 3.1 holds. Let \( \hat{\omega}_k = (\hat{x}^k_1, \hat{x}^k_2, \hat{x}^k_3, \hat{\lambda}_k) \) be generated by (10), let \( \omega_{k+1} \) and \( \alpha_k \) be defined as (11) and (12). Then for any \( \omega^* \in W^* \), it follows that

\[
\|\omega_{k+1} - \omega^*\|_2^2 \leq \|\omega_k - \omega^*\|_2^2 - \alpha_k \phi(\omega_k, \tilde{\omega}_k).
\]

**Proof.** Let

\[
\omega_{k+1} = \omega_k - \alpha M^T(\omega_k - \tilde{\omega}_k),
\]

then we have

\[
\begin{align*}
\|\omega_k - \omega^*\|_2^2 - \|\omega_{k+1} - \omega^*\|_2^2 &= \|\omega_k - \omega^*\|_2^2 - \|\omega_k - \omega^* - \alpha M^T(\omega_k - \tilde{\omega}_k)\|_2^2 \\
&= 2\alpha(\omega_k - \tilde{\omega}_k)^T M(\omega_k - \omega^*) - \alpha^2(\omega_k - \tilde{\omega}_k)^T M M^T(\omega_k - \tilde{\omega}_k) \\
&\geq 2\alpha \phi(\omega_k, \tilde{\omega}_k) - \alpha^2 \|M^T(\omega_k - \tilde{\omega}_k)\|_2^2.
\end{align*}
\]

Define

\[
\psi(\alpha) = 2\alpha \phi(\omega_k, \tilde{\omega}_k) - \alpha^2 \|M^T(\omega_k - \tilde{\omega}_k)\|_2^2.
\]

It is obvious that when

\[
\alpha = \frac{\phi(\omega_k, \tilde{\omega}_k)}{\|M^T(\omega_k - \tilde{\omega}_k)\|_2^2},
\]

\( \psi(\alpha) \) attains its maximum \( \|M^T(\omega_k - \tilde{\omega}_k)\|_2^2 \). We have

\[
\|\omega_{k+1} - \omega^*\|_2^2 \leq \|\omega_k - \omega^*\|_2^2 - \frac{\phi^2(\omega_k, \tilde{\omega}_k)}{\|M^T(\omega_k - \tilde{\omega}_k)\|_2^2},
\]

which completes the proof. 

**Corollary 1.** Suppose that Assumption 3.1 holds. Let \( \hat{\omega}_k = (\hat{x}^k_1, \hat{x}^k_2, \hat{x}^k_3, \hat{\lambda}_k) \) be generated by (10), then for any \( \omega^* \in W^* \), we have

1. The sequence \( \|\omega_k - \omega^*\|_2 \) is non-increasing.
2. \( \lim_{k \to +\infty} \phi(\omega_k, \tilde{\omega}_k) = 0 \) which implies \( \lim_{k \to +\infty} \|\omega_k - \tilde{\omega}_k\|_2 = 0 \).
3. The sequences \( \{\omega_k\} \) and \( \{\tilde{\omega}_k\} \) are both bounded.

In this section, we present the convergence result of the proposed algorithm.

**Theorem 3.4.** Suppose that Assumption 3.1 holds. Let \( \{\omega_k\} \) be the sequence generated by (10), then \( \{\omega_k\} \) converges to some \( \omega^* \in W^* \).

**Proof.** From Corollary 3.1, we know that \( \{\tilde{\omega}_k\} \) is bounded, hence it has at least one cluster point. Let \( \omega^* \) be the cluster point of \( \{\tilde{\omega}_k\} \). As a result, there exists a subsequence \( \{k_i\} \) such that

\[
\lim_{k_i \to +\infty} \tilde{\omega}_{k_i} = \omega^*.
\]

According to Corollary 3.1, we get

\[
\lim_{k \to +\infty} \|\omega_k - \omega^*\|_2 = 0,
\]

i.e.

\[
\lim_{k \to +\infty} \|\tilde{x}_1^k - \tilde{x}_1^k\| = \lim_{k \to +\infty} \|\tilde{x}_2^k - \tilde{x}_2^k\| = \lim_{k \to +\infty} \|\tilde{x}_3^k - \tilde{x}_3^k\| = \lim_{k \to +\infty} \|\lambda_k - \tilde{\lambda}_k\| = 0.
\]
Numerical experiments.

From the optimal condition of (15), we have

\[
\lim_{k \to +\infty} \begin{pmatrix}
  x_1 - \hat{x}_1^k \\
  x_2 - \hat{x}_2^k \\
  x_3 - \hat{x}_3^k \\
  \lambda_k - \lambda_k
\end{pmatrix}^T \begin{pmatrix}
  \nabla_{x_1} \mathcal{L}_\beta(\hat{x}_1^k, \hat{x}_2^k, \hat{x}_3^k, \hat{\lambda}_k) - A_1^T \hat{\lambda}_k \\
  \nabla_{x_2} \mathcal{L}_\beta(\hat{x}_1^k, \hat{x}_2^k, \hat{x}_3^k, \hat{\lambda}_k) - A_2^T \hat{\lambda}_k \\
  \nabla_{x_3} \mathcal{L}_\beta(\hat{x}_1^k, \hat{x}_2^k, \hat{x}_3^k, \hat{\lambda}_k) - A_3^T \hat{\lambda}_k \\
  A_1 \hat{x}_1^k + A_2 \hat{x}_2^k + A_3 \hat{x}_3^k - b + \frac{1}{\beta} (\lambda_k - \lambda_k)
\end{pmatrix} \geq 0
\]

(28)

for all \((x_1, x_2, x_3) \in \Omega\). It follows from (28) that

\[
\lim_{k_i \to +\infty} \begin{pmatrix}
  x_1 - \hat{x}_1^{k_i} \\
  x_2 - \hat{x}_2^{k_i} \\
  x_3 - \hat{x}_3^{k_i} \\
  \lambda_k - \lambda_{k_i}
\end{pmatrix}^T \begin{pmatrix}
  \nabla_{x_1} \mathcal{L}_\beta(\hat{x}_1^{k_i}, \hat{x}_2^{k_i}, \hat{x}_3^{k_i}, \hat{\lambda}_{k_i}) - A_1^T \hat{\lambda}_{k_i} \\
  \nabla_{x_2} \mathcal{L}_\beta(\hat{x}_1^{k_i}, \hat{x}_2^{k_i}, \hat{x}_3^{k_i}, \hat{\lambda}_{k_i}) - A_2^T \hat{\lambda}_{k_i} \\
  \nabla_{x_3} \mathcal{L}_\beta(\hat{x}_1^{k_i}, \hat{x}_2^{k_i}, \hat{x}_3^{k_i}, \hat{\lambda}_{k_i}) - A_3^T \hat{\lambda}_{k_i} \\
  A_1 \hat{x}_1^{k_i} + A_2 \hat{x}_2^{k_i} + A_3 \hat{x}_3^{k_i} - b + \frac{1}{\beta} (\lambda_k - \lambda_{k_i})
\end{pmatrix} \geq 0
\]

(29)

for all \((x_1, x_2, x_3) \in \Omega\). Consequently,

\[
\begin{pmatrix}
  x_1 - x_1^* \\
  x_2 - x_2^* \\
  x_3 - x_3^* \\
  \lambda_k - \lambda^*
\end{pmatrix}^T \begin{pmatrix}
  \nabla_{x_1} \mathcal{L}_\beta(x_1^*, x_2^*, x_3^*) - A_1^T \lambda^* \\
  \nabla_{x_2} \mathcal{L}_\beta(x_1^*, x_2^*, x_3^*) - A_2^T \lambda^* \\
  \nabla_{x_3} \mathcal{L}_\beta(x_1^*, x_2^*, x_3^*) - A_3^T \lambda^* \\
  A_1 x_1^* + A_2 x_2^* + A_3 x_3^* - b
\end{pmatrix} \geq 0, \quad \forall (x_1, x_2, x_3) \in \Omega,
\]

(30)

which implies \(\omega^* \in W^*\). From (26)-(27), for any \(\epsilon > 0\), there exists an integer \(K\) such that \(\|\omega_K - \tilde{\omega}_K\|_2 < \frac{\epsilon}{2}\) and \(\|\tilde{\omega}_K - \omega^*\|_2 < \frac{\epsilon}{2}\). According to Corollary 3.1, we know the sequence \(\|\omega_k - \omega^*\|_2\) is non-increasing. Therefore, for any \(k > K\),

\[
\|\omega_k - \omega^*\|_2 \leq \|\omega_k - \omega^*\|_2 \leq \|\omega_k - \tilde{\omega}_K\|_2 + \|\tilde{\omega}_K - \omega^*\|_2 \leq \epsilon.
\]

which implies that the sequence \(\{\omega_k\}\) converges \(\omega^* \in W^*\). \(\square\)

4. Numerical experiments. In the section, we report several numerical results to illustrate the validity of the multi-block proximal alternating direction method, which is denoted as PADM. We code our algorithm in MATLAB R2012b and all the tests are executed on a computer equipped with Inter(R) Core(TM)2 Duo CPU, 2.93 GHZ, 1.93GB RAM, running on Window XP system.

Example 1. Solve a linear system of equations. Chen [5] proposed that the classic alternating direction method of multiplier (ADMM) to solve a linear system of equations can not be extended to three or more blocks directly, and gave a counterexample that could be diverge. Next, we consider the linear system of equations

\[
A_1 x_1 + A_2 x_2 + A_3 x_3 = b,
\]

(31)

where \(A_1 \in R^{p \times m}, A_2 \in R^{p \times n}, A_3 \in R^{p \times l}\), and \(A_i (i = 1, 2, 3)\) follow the Gaussian distribution \(N(0, 1)\). We denote \(A = (A_1, A_2, A_3)\), and \(x = (x_1^T, x_2^T, x_3^T)^T\). \(x_0 \in R^{p \times (m+n+l)}\) is randomly generated vector whose every component follows the Gaussian distribution \(N(0, 1)\). We choose \(b = Ax_0\). The example is a special case for model (15). We compare our method with LADM [24], GADMM [17], ALM [26], BSUM-M [27]. The parameter setting of these mentioned methods refer to relevant literatures, especially, penalty parameter of all algorithms are same for the same problem. The rest problem can be done in the same manner. First, we set \(m = 50, n = 40, l = 60, p = 250\). The maximal number of iterations is 2000 and the stopping criterion is \(\|Ax - b\| \leq 10^{-10}\). Let \(Q = 1000, \beta = 0.8\). During each iteration, we use \(x = (x_1^T, x_2^T, x_3^T)^T\) generated randomly as the initial point to solve the
subproblem. Part (a) of Fig 1 describes the convergence precision of the algorithms. PADM takes a self-adaptive stepsize, and obtain relative high accuracy than other algorithms. When the problem size is enlarged, we find that our method is still competitive. Part (b) of Fig 1 shows the result when \( m = 200, n = 160, l = 240, p = 800 \).

\[
\text{Fig 1. Convergence precision of all algorithms, the error is given by } \|Ax - b\|.
\]

**Example 2. A generalized Nash equilibrium problem.** This example is a GNEP from [13], which consists of two players. One player controls a two-dimensional variable \( x = (x_1, x_2) \), and another player controls a one-dimensional variable \( y \in \mathbb{R} \). This problem has an infinite number of solutions given by \( \{(t, 11 - t, 8 - t) | t \in [0, 2] \} \). It is obvious that inequality constraints are active at all the solutions of the problem. Hence, it has same solutions as the problem with equality constraint. Then the GNEP is equivalent to the following problem with three players: Each player controls a one-dimensional variable, the first player and the second player have the same objective function.

\[
\begin{align*}
\min \quad & \theta_1(x_1, x_2, x_3) = x_1^2 + x_1 x_2 + x_2^2 + (x_1 + x_2) x_3 - 25 x_1 - 38 x_2 \\
\text{s.t} \quad & x_1 + 2 x_2 - x_3 = 14, \\
& 3 x_1 + 2 x_2 + x_3 = 30, \\
& x_1 \geq 0, x_2 \geq 0, i = 1, 2.
\end{align*}
\]

and

\[
\begin{align*}
\min \quad & \theta_3(x_1, x_2, x_3) = x_2^2 + (x_1 + x_2) x_3 - 25 x_3 \\
\text{s.t} \quad & x_1 + 2 x_2 - x_3 = 14, \\
& 3 x_1 + 2 x_2 + x_3 = 30, \\
& x_1 \geq 0, x_2 \geq 0.
\end{align*}
\]

Let

\[
\nabla_x \theta(x_1, x_2, x_3) = \begin{pmatrix}
\nabla_{x_1} \theta_1(x_1, x_2, x_3) \\
\nabla_{x_2} \theta_2(x_1, x_2, x_3) \\
\nabla_{x_3} \theta_3(x_1, x_2, x_3)
\end{pmatrix} = H x + g,
\]

where

\[
H = \begin{pmatrix}
2 & 1 & 1 \\
1 & 2 & 1 \\
1 & 1 & 2
\end{pmatrix}, \quad g = \begin{pmatrix}
-25 \\
-38 \\
-25
\end{pmatrix}.
\]
In this experiment, we set $Q = 10$, $\beta = 0.8$, the maximal number of iterations is 100 and the stopping criterion is $\|\omega_k - \tilde{\omega}_k\| \leq 10^{-10}$. We use self-adaptive projection method [23] to solve the subproblems (10). Regardless of the starting point, we find that PADM always gets a solution $(0, 11, 8)$ and $\|\omega_k - \tilde{\omega}_k\| = 9.0091e - 09$. From Fig.2 (a), we see that each player can find optimal solution respectively. From the Fig.2(b), we find that PADM has better results than ALM [35] and GADMM [17], while BSUM-M [27] performs better. The convergence rate of the BSUM-M [27] is faster than other methods and its accuracy is also higher than other methods. We observe that when the stepsize $\alpha_k$ is fixed, PADM still obtains a good result. Fig 3 shows the results when $\alpha_k = 0.2$ and $\beta = 0.3$. It is necessary to note that choosing a proper stepsize is not easy work for BSUM-M, which reveals the significance of self-adaptive stepsize.

**Example 3. The basis pursuit problem.** In this example, we consider the following nonsmooth problem

$$\min_x \|x\|_1,$$

$$s.t. \ Ax = b. \ x \in \mathcal{X}$$

(32)
The basis pursuit problem is deemed to recover the sparsity vector $x$ from a small number of observations $b$ applied in compressive sensing effectively. We divide the sparsity vector $x$ into $N$-block, i.e., $x = (x_1, x_2, \ldots, x_N)^T, x_i \in \mathbb{R}^{n_i}, \text{i}=1,2,\ldots, N$. Therefore, (32) can be rewritten as

$$
\min_x \sum_{i=1}^{N} |x_i|
onumber
$$

$$
s.t. \sum_{i=1}^{N} A_i x_i = b, \quad x_i \in \mathcal{X}_i, \quad \text{i}=1,2,\ldots, N. \quad (33)
$$

We generated $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ similarly to the Example 1. Let $m = 250, n = 1000$. We randomly generate matrix $A$ following the Gaussian distribution $\mathcal{N}(0, 1)$, and orthonormalize its rows. Then we generate a sparse vector $x_0 \in \mathbb{R}^{n}$ with 25 nonzero entries, each component of $x_0$ follows Gaussian distribution $\mathcal{N}(0, 1)$. The observed vector $b$ is generated by $b = Ax_0 + \xi$, where $\xi \sim \mathcal{N}(0, 10^{-3}I)$. The maximal number of iterations is set as 500, the stopping criterion is as well as Example 2, and $\beta = 0.05, \varepsilon = 10^{-15}$. We can clearly see the variation of the error with the iteration increased in Fig 4, and PADM has a comparison with FISTA [2], PALM [38], LADM [24], GADMM [17], ALM1 [26], BSUM-M [27]. The results show that PADM obtained superior performance over all other algorithms.

**Example 4. The constrained LASSO.** We consider the constrained LASSO (CLASSO) problem [30] as follows:

$$
\min_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \nonumber
$$

$$
s.t. \quad Cx = d, x \geq 0, \quad (34)
$$

where $A \in \mathbb{R}^{p \times n}$ is a feature matrix, $b \in \mathbb{R}^n$ is an observed vector. The $l_1$ norm is defined as $\|x\|_1 = \sum_{i=1}^{n} |x_i|$, was known to promote the sparsity of the solution. Many commonly applied statistical problem, such as the fused lasso, generalized lasso, monotone curve estimation etc., can be expressed as special cases of (34). The problem has the same structure as model (1).
In this experiment, $A \in \mathbb{R}^{p \times n}$, $C \in \mathbb{R}^{m \times n}$ and a sparse vector $x_0 \in \mathbb{R}^n$ with 12 nonzero entries is randomly generated whose each element follows the Gaussian distribution $\mathcal{N}(0, 1)$. Let $p = 120$, $n = 1200$, $m = 20$. We get vector $b = Ax_0 + \gamma$, $d = Cx_0$, where each component of $\gamma \sim \mathcal{N}(0, 10^{-3}I)$, and choose the model parameter $\lambda = 0.1$. The maximal number of iterations is set as 100, and the stopping criteria $\varepsilon = 10^{-15}$. Under this setting, PADM compares with ALM [35] and GADMM [17], BSUM-M [27]. From the Fig.5, we know that the result obtained by PADM is far better than other algorithms. The comparative analysis suggests that PADM is efficient to solve model (1).

5. **Conclusion.** In this paper, we proposed a multi-block proximal alternating direction method to solve the linearly constrained convex minimization model with an objective function which is the sum of several separable functions and a coupled quadratic function. Under the mild condition that $H$ is symmetric positive semidefinite, we show the global convergence of the algorithm. The numerical experiments suggested that our method is effective and competitive while comparing with other methods.

**Acknowledgments.** The authors would like to thank the reviewers for their valuable suggestions towards the improvement of the paper.

**REFERENCES**

[1] A. Agarwal, S. Negahban and M. J. Wainwright, Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions, *Ann. Appl. Stat.*, **40** (2012), 1171–1197.

[2] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imag. Sci.*, **2** (2009), 183–202.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *FnT Mach. Learn.*, **3** (2010), 1–122.

[4] X. Cai, D. Han and X. Yuan, On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function, *Comput. Optim. Appl.*, **66** (2017), 39–73.

[5] C. Chen, B. He, X. Yuan and Y. Ye, The direct extension of ADMM for Muti-block convex minimization problems is not necessarily convergent, *Math. Program.*, **155** (2016), 57–79.
[6] C. Chen, M. Li, X. Liu and Y. Ye, Extended ADMM and BCD for nonseparable convex minimization models with quadratic coupling terms: Convergence analysis and insights, *Mathematics*, 65 (2017), 1231–1249.

[7] C. Chen, Y. Shen and Y. You, On the convergence analysis of the alternating direction method of multipliers with three blocks, *Abstr. Appl. Anal.*, 2013 (2013), Art. ID 183961, 7 pp.

[8] Y. Cui, X. Li, D. Sun and K. C. Toh, On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions, *J. Optim. Theory Appl.*, 169 (2016), 1013–1041.

[9] D. Davis and W. Yin, Convergence rate analysis of several splitting schemes, *UCLA CAM Report*, 2014, 14–51.

[10] W. Deng and W. Yin, On the global and linear convergence of the generalized alternating direction method of multipliers, *J Sci. Comput.*, 66 (2016), 889–916.

[11] J. Eckstein and M. Fukushima, Some reformulation and applications of the alternating direction method of multipliers, *Scale Optim.*, 1994, 115–134.

[12] J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, *Math. Program.*, 55 (1992), 293–318.

[13] F. Facchinei and C. Kanzow, Penalty methods for the solution of generalized Nash equilibrium problems, *SIAM J. Control. Optim.*, 20 (2010), 2228–2253.

[14] C. Feng, H. Xu and B. Li, An Alternating direction method approach to cloud traffic management, *IEEE T. Parall. Distr.*, 28 (2017), 2145–2158.

[15] M. Fortin and R. Glowinski, Augmented Lagrangian methods: Applications to the numerical solution of boundary value problems, *Stud. Math. Appl.*, 15 (1983), xix+340 pp.

[16] D. Gabay and B. Mercier, A dual algorithm for the solution of non-linear variational problems via finite element approximations, *Comput. Optim. Appl.*, 2 (1976), 17–40.

[17] X. Gao and S. Zhang, First-Order algorithms for convex optimization with nonseparable objective and coupled constraints, *J Oper. Res. Soc. China.*, 5 (2017), 131–159.

[18] R. Glowinski and A. Marroco, Sur l’approximation, par éléments finis d’ordre un, et la résolution, par penalisation-dualité, d’une classe de problèmes de dirichlet non linéaires, *J Equine. Vet. Sci.*, 9 (1975), 41–76.

[19] D. Han, X. Yuan and W. Zhang, An augmented-Lagrangian-based parallel splitting method for separable convex minimization with applications to image processing, *Math. Comput.*, 83 (2014), 2263–2291.

[20] D. Han and X. Yuan, A note on the alternating direction method of multipliers, *J. Optim. Theory Appl.*, 155 (2012), 227–238.

[21] D. Han, X. Yuan, W. Zhang and X. Cai, An ADM-based splitting method for separable convex programming, *Comput. Optim. Appl.*, 54 (2013), 343–369.

[22] B. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Comput. Optim. Appl.*, 42 (2009), 195–212.

[23] B. He, H. Yang, Q. Meng and D. Han, Modified Goldstein-Levitin-Polyak projection method for asymmetric strongly monotone variational inequalities, *J. Optim. Theory Appl.*, 112 (2002), 129–143.

[24] B. He, M. Tao, and X. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, *SIAM J. Optim.*, 22 (2012), 313–340.

[25] B. He, M. Tao and X. Yuan, A splitting method for separable convex programming, *IMA J Numer. Anal.*, 35 (2015), 394–426.

[26] B. He, L. Hou and X. Yuan, On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming, *SIAM J. Optim.*, 25 (2015), 2274–2312.

[27] M. Hong, T. Chang, X. Wang, M. Razaviyayn, S. Ma and Z. Luo, A block successive upper bound minimization method of multipliers for linearly constrained convex optimization, *Mathematics*, 2014.

[28] M. Hong, Z. Luo and M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM J. Optim.*, 26 (2016), 337–364.

[29] M. Hong and Z. Luo, On the linear convergence of the alternating direction method of multipliers, *Math. Program.*, 162 (2017), 165–199.

[30] G. James, C. Paulson and P. Rusmevichientong, *The Constrained Lasso*, Technical report, University of Southern California, 2013.

[31] X. Li, D. Sun and K. C. Toh, A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions, *Math. Program.*, 155 (2016), 333–373.
[32] T. Lin, S. Ma and S. Zhang, On the global linear convergence of the ADMM with multi-block variables, *SIAM J. Optim.*, 25 (2015), 1478–1497.

[33] J. F. Mota, J. M. Xavier, P. M. Aguiar and M. Puschel, Distributed optimization with local domains: Application in MPF and network flows, *IEEE T. Automat. Contr.*, 60 (2015), 2004–2009.

[34] Y. Peng, A. Ganesh, J. Wright, W. Xu and Y. Ma, Robust alignment by sparse and low-rank decomposition for linearly correlated images, *IEEE T. Pattern. Anal.*, 34 (2012), 2233–2246.

[35] R. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Math. Oper. Res.*, 1 (1976), 97–116.

[36] D. Sun, K. C. Toh and L. Yang, A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with 4-block constraints, *SIAM J. Optim.*, 25 (2015), 882–915.

[37] L. Xu and D. Han, A proximal alternating direction method for weakly coupled variational inequalities, *Pacific J. Optim.*, 9 (2013), 155–166.

[38] J. Yang and Y. Zhang, Alternating direction algorithms for $\ell_1$-Problems in compressive sensing, *SIAM J. Sci. Comput.*, 33 (2011), 250–278.

[39] X. Yuan, An improved proximal alternating directions method for monotone variational inequalities with separable structure, *Comput. Optim. Appl.*, 49 (2011), 17–29.

Received August 2016; revised December 2017.

E-mail address: lfx_pjnu7math@sina.com
E-mail address: xulingling@njnu.edu.cn
E-mail address: sunyuehong@njnu.edu.cn
E-mail address: handeren@njnu.edu.cn