The proximal-proximal gradient algorithm

Ting Kei Pong *

May 5, 2013

Abstract

We consider the problem of minimizing a convex objective which is the sum of a smooth part, with Lipschitz continuous gradient, and a nonsmooth part. Inspired by various applications, we focus on the case when the nonsmooth part is a composition of a proper closed convex function $P$ and a nonzero affine map, with the proximal mappings of $\tau P$, $\tau > 0$, easy to compute. In this case, a direct application of the widely used proximal gradient algorithm does not necessarily lead to easy subproblems. In view of this, we propose a new algorithm, the proximal-proximal gradient algorithm, which admits easy subproblems. Our algorithm reduces to the proximal gradient algorithm if the affine map is just the identity map, and it is equivalent to applying a variant of the alternating minimization algorithm [34] to the dual problem. Moreover, it is closely related to inexact proximal gradient algorithms [28, 32]. We show that the whole sequence generated from the algorithm converges to an optimal solution. We also establish an upper bound on iteration complexity. Our numerical experiments on the stochastic realization problem and the logistic fused lasso problem suggest that the algorithm performs reasonably well on large-scale instances.

1 Introduction

We consider the following optimization problem

$$
\nu_{\text{opt}} := \min_z F(z) := h(z) + P(Mz - b),
$$

where $h$ is a convex differentiable function with a Lipschitz continuous gradient whose Lipschitz continuity modulus is bounded by $L$, $P$ is a proper closed convex function, $M$ is a nonzero linear map between two finite dimensional Hilbert spaces $Z$ and $Y$, and $b$ is a given element in $Y$. We assume throughout that (1) has an optimal solution. We also assume the following constraint qualification to rule out degenerate instances:

$$(\text{Range}(M) - b) \cap \text{ri}(\text{dom} P) \neq \emptyset;$$

see Section 1.1 for notations and definitions. This condition is trivially satisfied if $P$ is continuous or $M$ is surjective. Furthermore, we assume that the proximal mapping of $P$, i.e., the quantity

$$\text{prox}_P(y) := \arg \min_u \left\{ P(u) + \frac{1}{2} \|u - y\|^2 \right\}$$

is easy to compute for any given $y$, so is that of $\tau P$ for any $\tau > 0$;

*Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada, N2L 3G1. E-mail: ptingkei@uwaterloo.ca. This author was supported by research grants from AFOSR and NSERC.
Problem (1) arises from various applications, with $h$ usually the loss function, $P$ the regularization function and $Mz - b$ modeling some priors known about the desired solution. One example arises from the maximum a posteriori probability analysis approach in signal processing [4], where $b = 0$, $P$ is the $\ell_1$ norm and $M$ is the so-called analyzing operator, with a higher value of $P(Mz)$ for a less probable signal. Another example arises from the system identification or system realization problems, where $P$ is the nuclear norm of a matrix (i.e., sum of singular values of a matrix), $b = 0$ and $Mz$ is a block Hankel matrix of suitable size; see, for example, [9, 17, 18, 19, 37]. Finally, we also mention the fused lasso logistic regression problem introduced recently in [21] in which $h$ is the smooth logistic loss function, $b = 0$, $P$ is the $\ell_1$ norm and $M$ is a suitably defined linear map. We refer the readers to [21, Section 5] for more details about this particular problem. Notice that in all the above applications, the proximal mapping of $P$ and those of $\tau P$, $\tau > 0$ are computable via a closed form formula; see, for example, [33, Section 9.2].

Since $h$ is differentiable with Lipschitz continuous gradient and $P$ is nonsmooth, it is natural to consider a proximal gradient algorithm [10] for solving (1), where one updates

$$z^{t+1} = \arg \min_z \left\{ \langle \nabla h(z^t), z \rangle + P(Mz - b) + \frac{L}{2} \| z - z^t \|^2 \right\}. \tag{3}$$

For such an algorithm, it is well-known that $F(z^t) - F(z^*) = O \left( \frac{1}{t} \right)$; see, for example, [25, Theorem 2.1.14]. Furthermore, the whole sequence generated is convergent to an optimal solution of (1); see, for example, [24, Section 2]. This simple algorithm, together with its accelerated versions using Nesterov extrapolation techniques [24, 25, 26, 27], has recently been studied extensively in the literature; see, for example, [1, 2, 9, 16, 20, 22, 29, 31, 35, 36]. While the proximal gradient algorithm has nice convergence property, for the algorithm to be efficient, the optimization problem in (3) should be easy to solve. This problem is the same as computing the proximal mapping of $\frac{1}{L} P(M \cdot - b)$. While the proximal mapping of $\frac{1}{L} P$ is easy to compute, the proximal mapping of the composition $\frac{1}{L} P(M \cdot - b)$, however, is in general not trivial to compute.

To get around this difficulty, one natural way is to “decouple” the function $P$ and the affine map. One possible way is to reformulate the problem as

$$\min_{z, u} \ h(z) + P(u) \quad \text{s.t.} \quad u = Mz - b, \tag{4}$$

and apply a suitable algorithm, for example, the alternating direction method of multipliers (ADMM); see, for example, [5, 6, 8, 11, 12, 13]. This algorithm consists of the following updates

$$\begin{align*}
    z^{t+1} &\in \text{Arg min}_z \left\{ h(z) + \langle y^t, Mz \rangle + \frac{\beta}{2} \| u^t - Mz + b \|^2 \right\}, \\
    u^{t+1} &\in \text{Arg min}_u \left\{ P(u) - \langle y^t, u \rangle + \frac{\beta}{2} \| u - Mz^{t+1} + b \|^2 \right\}, \\
y^{t+1} &= y^t - \gamma \beta (u^{t+1} - Mz^{t+1} + b),
\end{align*}$$

where $\beta > 0$ and $\gamma \in (0, \frac{\sqrt{\beta}}{2})$. While the second subproblem is just a computation of the proximal mapping of $\frac{1}{\beta} P$, the first subproblem involving $h$ is in general hard to solve. Recent variants of this method have proximal terms added to the objectives for the $z$-update and $u$-update, in order to reduce the subproblem to a computation of the proximal mappings of $\frac{1}{\beta} h$ and $\frac{1}{\beta} P$, respectively; see, for example, [39, 41]. However, the proximal mapping of $\frac{1}{\beta} h$ can still be difficult to compute. In the recent work [21], a variant of the above ADMM was proposed to solve (1) so that each subproblem has a simple closed form solution. For this method, it was only shown that any cluster point of the average of the sequence $\{(z^t, u^t)\}$ generated minimizes a Lagrangian function of (1).
Another possible way to “decouple” the function $P$ and the affine map is to reformulate the problem into a convex-concave optimization problem. In this approach, one first rewrite (1) as a convex-concave optimization problem as follows:

$$\min_z \max_y \left\{ h(z) + \langle y, Mz - b \rangle - P^*(y) \right\},$$

(5)

where $P^*$ is the convex conjugate of $P$. One can then apply a suitable algorithm, for example, the variant of Tseng’s modified forward backward splitting (MFBS) method proposed in [23, Section 5], which consists of the following updates in each iteration

\[
\begin{align*}
    v^t &= \arg\min_y \left\{ P^*(y) + \langle b - Mz^t, y \rangle + \frac{L_M}{2\sigma} \|y - y^t\|^2 \right\}, \\
    u^t &= z^t - \frac{\sigma}{L_M} (\nabla h(z^t) + M^* y^t), \\
    y^{t+1} &= v^t - \frac{\sigma}{L_M} (Mz^t - Mu^t), \\
    z^{t+1} &= u^t - \frac{\sigma}{L_M} (\nabla h(u^t) + M^* v^t - \nabla h(z^t) - M^* y^t),
\end{align*}
\]

(6)

where $\sigma \in (0, 1)$ and $L_M$ is an upper bound of the Lipschitz continuity modulus of the function $G(z, y) = \left(\nabla h(z) + M^* y \right)$, and $M^*$ is the adjoint linear map of $M$. The subproblem in (6) can be shown to be equivalent to a computation of the proximal mapping of $\frac{\sigma}{L_M} P$, and it can be further shown using [23, Proposition 2.3] and [23, Theorem 2.4] that the sequence $\{(u^t, v^t)\}$ converges to a saddle point of (5).

In this paper, we propose a new algorithm, called the proximal-proximal gradient algorithm, to solve (1), which has nice convergence property and admits easy subproblems in each iteration like the MFBS method mentioned above. Our algorithm is obtained by applying a variant of the alternating minimization algorithm [34] (AMA) to the dual problem of (1). It reduces to the proximal gradient algorithm if $M$ is the identity map and $b = 0$, and is closely related to inexact proximal gradient algorithms [28, 32]. For this new algorithm, we prove that the whole sequence generated converges to an optimal solution of (1) and establish an upper bound on iteration complexity. We also test our algorithm on the stochastic realization problem (see, for example, [18]) against the primal ADMM2 in [9] and the MFBS method in [23], as well as on the fused lasso logistic regression problem (see [21]) against the MFBS method in [23]. Our computational results show that our algorithm works reasonably well on large-scale instances and usually outperforms the MFBS method.

The rest of the paper is organized as follows. We introduce notations used throughout the paper in Section 1.1 and motivate our algorithm in Section 2. In Section 3, we propose a proximal alternating minimization algorithm and establish its global convergence. In Section 4, we present our proximal-proximal gradient algorithm and discuss its various properties, including convergence and its relationship with inexact proximal gradient algorithms. Numerical results are reported in Section 5. Finally, we give concluding remarks in Section 6.

1.1 Notations

In this paper, scripted letters $X$, $Y$ and $Z$ denote finite dimensional Hilbert spaces. Abusing notations, we use $\langle \cdot, \cdot \rangle$ to denote the inner product and $\| \cdot \|$ to denote the norm induced from the inner product on each of the spaces. Linear maps between finite dimensional Hilbert spaces are denoted by scripted letters (other than $X$, $Y$ and $Z$). For a linear map $A$, $A^*$ denotes the adjoint linear map with respect to the inner product, $\text{Range}(A)$ denotes the image of $A$ and $\|A\|$ is the induced operator norm of $A$. A linear self-map $T : Z \to Z$ is called positive semidefinite.
(resp., positive definite) if $\mathcal{T} = \mathcal{T}^*$ and $\langle z, \mathcal{T} z \rangle \geq 0$ (resp., $\langle z, \mathcal{T} z \rangle > 0$) for all nonzero $z \in Z$. We let $\mathcal{T} \succeq 0$ (resp., $\mathcal{T} > 0$) denote $\mathcal{T}$ is a positive semidefinite (resp., positive definite) linear map. For a positive semidefinite linear map $\mathcal{T}$, $\| \cdot \|_\mathcal{T}$ denotes its induced seminorm defined by $\|z\|_\mathcal{T} := \sqrt{\langle z, \mathcal{T} z \rangle}$ for all $z \in Z$. We denote the identity map by $\mathcal{I}$.

For a closed convex function $f : Z \to (-\infty, \infty]$, the domain of $f$ is a convex set and is defined by $\text{dom } f := \{z \in Z : f(z) < \infty\}$. A closed convex function $f$ is called proper if $\text{dom } f \neq \emptyset$. The set of subdifferentials of $f$ at a point $z \in Z$, denoted by $\partial f(z)$, is defined by

$$\partial f(z) := \{v \in Z : f(u) - f(z) \geq \langle v, u - z \rangle \forall u \in Z\}.$$ 

For a proper closed convex function $f : Z \to (-\infty, \infty]$, the convex conjugate $f^*$ of $f$ is the proper closed convex function defined by

$$f^*(z) := \sup_u \{\langle z, u \rangle - f(u)\}.$$ 

It is well-known [30, Theorem 12.2] that

$$f^{**} = f. \quad (7)$$ 

The proximal mapping of $f$ is defined by

$$\text{prox}_f(z) := \arg \min_u \left\{ f(u) + \frac{1}{2} \|u - z\|^2 \right\},$$

where $\arg \min$ denotes the minimizer. Recall from [30, Theorem 31.5] that the minimizer of the above optimization problem always exists and is unique, hence the proximal mapping is well-defined. Moreover, from the same theorem, we have the following relation:

$$\text{prox}_f(z) + \text{prox}_{f^*}(z) = z \quad \forall z \in Z. \quad (8)$$

Finally, we use $\text{ri}(C)$ to denote the relative interior of a convex set $C$.

## 2 Motivations

Recall that the main difficulty for the efficient implementation of the proximal gradient algorithm (3) lies in the fact that the proximal mapping of $\tau P(M \cdot -b)$, $\tau > 0$, is not necessarily easy to compute. In this section, we will try to look at the proximal gradient algorithm (3) from a dual perspective. In particular, we revisit the well-known fact that the proximal gradient algorithm is equivalent to applying the AMA to the dual of (1), a result established in [34, Section 2].

To describe the equivalence, we note from (2) and [30, Theorem 31.2] that

$$v_{\text{opt}} = \min_z h(z) + P(Mz - b) = \max_y -h^*(-M^*y) - P^*(y) - \langle b, y \rangle,$$

where $h^*$ and $P^*$ are the convex conjugate of $h$ and $P$ respectively. Thus, the dual of (1) is equivalent to the following problem:

$$\min_{x,y} h^*(x) + P^*(y) + \langle b, y \rangle \quad \text{s.t. } x + M^*y = 0. \quad (9)$$

Since $\nabla h$ is Lipschitz continuous with modulus bounded by $L$, the function $h^*$ is strongly convex with strong convexity modulus at least $\frac{1}{L}$. Hence, one can apply the AMA in [34, Section 4] to
Then it is easy to see that

\[
\begin{align*}
    x^{t+1} &= \arg\min_x \left\{ h^*(x) - \langle z^t, x \rangle \right\}, \\
y^{t+1} &\in \operatorname{Arg\,min}_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \| x^{t+1} + \mathcal{M}^* y \|^2 \right\}, \\
z^{t+1} &= z^t - \beta (x^{t+1} + \mathcal{M}^* y^{t+1}).
\end{align*}
\] (10)

This algorithm is known to converge if \( \beta \in (0, \frac{1}{L}) \), assuming that the iterates are well-defined; see [34 Proposition 3] for the precise assumptions. Moreover, with \( \beta = \frac{1}{L} \), it can be shown (see [34 Section 2]) that this algorithm is equivalent to the proximal gradient algorithm [33], in the sense that the \( z \)-iterates generated are the same. To be precise, we include the simple arguments below for completeness.

Indeed, from the first subproblem of (10) we see that

\[
    x^{t+1} = \arg\min_x \left\{ h^*(x) - \langle z^t, x \rangle \right\} \iff z^t \in \partial h^*(x^{t+1}) \iff x^{t+1} = \nabla h(z^t),
\] (11)

where the last equivalence follows from [33 Theorem 23.5]. On the other hand, from the second and third subproblem of (10), we see that

\[
\begin{align*}
y^{t+1} &\in \operatorname{Arg\,min}_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \| x^{t+1} + \mathcal{M}^* y \|^2 \right\} \\
\iff &\mathcal{M} z^{t+1} - b \in \partial P^*(y^{t+1}) \iff y^{t+1} \in \partial P(\mathcal{M} z^{t+1} - b) \\
\implies &\mathcal{M}^* y^{t+1} \in \mathcal{M}^* \partial P(\mathcal{M} z^{t+1} - b) \implies \mathcal{M}^* y^{t+1} \in \partial P(\mathcal{M} \cdot - b)(z^{t+1}),
\end{align*}
\] (12)

where the second equivalence follows from [33 Theorem 23.5] and the last implication follows from [33 Theorem 23.9]. From these it is easy to verify that \( \{z^t\} \) generated from the AMA satisfies [33]. Conversely, if \( \{z^t\} \) is generated according to [33], then from the first-order optimality condition, [2] and the subdifferential calculus rules [33 Theorem 23.8], [33 Theorem 23.9], we have

\[
0 \in \nabla h(z^t) + \mathcal{M}^* \partial P(\mathcal{M} z^{t+1} - b) + L(z^{t+1} - z^t).
\]

Define \( x^{t+1} = \nabla h(z^t) \) and let \( y^{t+1} \in \partial P(\mathcal{M} z^{t+1} - b) \) so that \( 0 = \nabla h(z^t) + \mathcal{M}^* y^{t+1} + L(z^{t+1} - z^t) \). Then it is easy to see that

\[
z^{t+1} = z^t - \frac{1}{L} (x^{t+1} + \mathcal{M}^* y^{t+1}).
\]

Using these relations, \( \beta = \frac{1}{L} \) and the equivalences in [11] and [12], we observe that the sequence \( \{(x^t, y^t, z^t)\} \) are the same iterates as generated from the AMA.

Since the AMA is in some sense equivalent to the proximal gradient algorithm, one should expect that the second subproblem involving \( P^* \) is no easier than [33]. However, we do gain some insights from this dual perspective. In the next section, borrowing the idea of adding a proximal term which was recently discussed in [11] in the settings of the ADMM, we develop a proximal alternating minimization algorithm which modifies the \( y \)-update in [10]. The resulting algorithm is then applied to [11] in Section 4 to give our proximal-proximal gradient algorithm.

### 3 A proximal alternating minimization algorithm

In this section, we consider a variant of the alternating minimization algorithm (AMA) proposed in [33 Section 4]. This method aims at solving optimization problems in the following form:

\[
\begin{align*}
    \min_{x} \quad & f(x) + g(y) \\
    \text{s.t.} \quad & Ax + By = c,
\end{align*}
\] (13)
where $A : X \to Z$ and $B : Y \to Z$ are linear maps on the finite dimensional Hilbert spaces $X$, $Y$ and $Z$, $c \in \mathbb{Z}$, $f : X \to (-\infty, +\infty]$ and $g : Y \to (-\infty, +\infty]$ are proper closed convex functions, with linear maps $\Sigma_f \succ 0$ and $\Sigma_g \succeq 0$ such that for any $u_1 \in \partial f(x_1)$ and $u_2 \in \partial f(x_2)$, we have

$$\langle u_1 - u_2, x_1 - x_2 \rangle \geq \| x_1 - x_2 \|^2_{\Sigma_f};$$

while for any $v_1 \in \partial g(y_1)$ and $v_2 \in \partial g(y_2)$, we have

$$\langle v_1 - v_2, y_1 - y_2 \rangle \geq \| y_1 - y_2 \|^2_{\Sigma_g},$$

where $\| \cdot \|_{\mathcal{T}}$ is the semi-norm induced by a positive semidefinite linear map $\mathcal{T}$. We add a proximal term in the second subproblem of the AMA and obtain the following so-called proximal AMA for solving the above problem. It reduces to the original AMA if $\mathcal{T} = 0$.

**Proximal AMA for (13)**

**Step 0.** Input $(y^0, z^0) \in Y \times Z$.

**Step 1.** Set

\[
\begin{align*}
x^{t+1} &= \arg \min_x \left\{ f(x) - \langle z^t, Ax \rangle \right\}, \\
y^{t+1} &= \arg \min_y \left\{ g(y) - \langle z^t, By \rangle + \frac{\beta}{2} \| Ax^{t+1} + By - c \|^2 + \frac{1}{2} \| y - y^t \|^2_{\mathcal{T}} \right\}, \\
z^{t+1} &= z^t - \beta (Ax^{t+1} + By^{t+1} - c),
\end{align*}
\]

where $\beta > 0$ and $\mathcal{T}$ is a positive semidefinite linear map.

**Step 2.** If a termination criterion is not met, go to Step 1.

When we apply the proximal AMA, we should expect $\min_x f(x) + \langle a, x \rangle$ to have an easy/closed form solution for any given $a$ so that the first subproblem in (16) is easy to solve. The motivation for introducing the proximal term in the second subproblem is to simplify the subproblem for the $y$-update by reducing it to a computation of the proximal mapping of $\frac{\beta}{2} g$, as discussed recently in [41] in the settings of ADMM. This would be particularly useful when the proximal mapping of $\frac{\beta}{2} g$ is easy to compute. Moreover, as discussed in [9] Appendix B], $\mathcal{T}$ should be chosen as “small” as possible in practice, and thus preferably not positive definite.

In the next theorem, we establish convergence of the above proximal AMA, under suitable assumptions. The tools used in the proof are quite standard and is largely based on [8] for the convergence of the alternating direction method of multipliers (ADMM), a closely related algorithm. See also [11] Section 6], [34] Appendix A], [38] and [9] Theorem B.1]. We also note that the convergence of AMA without proximal term (i.e., $\mathcal{T} = 0$) and with a varying step size $\beta$ was established in [34] Proposition 3], under slightly different assumptions from below.

Before stating the theorem, we make the following assumption:

**A1.** There exist $(\bar{x}, \bar{y}) \in X \times Y$ and $\bar{z} \in Z$ such that

$$A^* \bar{z} \in \partial f(\bar{x}), \quad B^* \bar{z} \in \partial g(\bar{y}), \quad A\bar{x} + B\bar{y} - c = 0.$$  \hspace{1cm} (17)

Note that for any $(\bar{x}, \bar{y}) \in X \times Y$ and $\bar{z} \in Z$ satisfying (17), $(\bar{x}, \bar{y})$ is an optimal solution to (13) and $\bar{z}$ is an optimal solution to the dual. Moreover, under a suitable constraint qualification, optimal solutions to the dual problem of (13) exist (see [30] Corollary 28.2.2)), with (17) being satisfied by
any optimal solution \((\bar{x}, \bar{y})\) to problem \((13)\) (see \([20\text{ Corollary }28.3.1]\)) and any optimal solution \(\tilde{z}\) to the dual problem of \((13)\) (see \([20\text{ Corollary }28.4.1]\)].

**Theorem 1.** Suppose that \(A_1\) holds and let \(\{(x^t, y^t, z^t)\}\) be generated from the proximal AMA. Suppose further that \(\Sigma_g + T + \beta B^* B > 0\) and \(2\Sigma_f - \beta A^* A > 0\). Then \(\{(x^t, y^t, z^t)\}\) is convergent and the limit satisfies \((17)\). In particular, \(\{(x^t, y^t)\}\) converges to an optimal solution to \((13)\) and \({z^t}\) converges to an optimal solution to the dual problem of \((13)\).

**Proof.** Notice that under our assumptions, the first and second subproblems in \((13)\) have strongly convex objectives. Hence the iterates are all well-defined. To proceed with the proof, we first rewrite the iterates of the proximal AMA as follows:

\[
\begin{align*}
0 \in \partial f(x^{t+1}) - A^* z^t, \\
0 \in \partial g(y^{t+1}) - B^* (z^t - \beta (Ax^{t+1} + By^{t+1} - c)) + T(y^{t+1} - y^t), \\
z^{t+1} = z^t - \beta (Ax^{t+1} + By^{t+1} - c).
\end{align*}
\]

(18)

For notational simplicity, we fix any \((\bar{x}, \bar{y}, \tilde{z})\) satisfying \((17)\) and write \(w_e := w^t - \bar{w}\), where \(w\) represents \(x\), \(y\) and \(z\) respectively. Using \((13)\), \((17)\), \((18)\) and \((19)\), we have

\[
\langle A^* e, x_e^{t+1} \rangle \geq \|x_e^{t+1}\|^2_{\Sigma_f}, \quad \langle B^* (e - \beta (Ax^{t+1} + By^{t+1})) - T(y^{t+1} - y^t), y_e^{t+1} \rangle \geq \|y_e^{t+1}\|^2_{\Sigma_g}.
\]

(19)

Summing these two inequalities and rearranging terms, we get

\[
\langle z_e^{t+1} + By_e^{t+1} - \beta (Ax^{t+1} + By^{t+1}), y_e^{t+1} - y^t \rangle \geq \|x_e^{t+1}\|^2_{\Sigma_f} + \|y_e^{t+1}\|^2_{\Sigma_g}.
\]

Substituting \(Ax^{t+1} + By^{t+1} = \beta (z^t - z^{t+1})\) in the first term on the left, we obtain further that

\[
\frac{1}{\beta} \langle z_e^t, z^t - z^{t+1} \rangle - \beta \langle Ax^{t+1} + By^{t+1}, y^{t+1} - y^t \rangle \geq \|x_e^{t+1}\|^2_{\Sigma_f} + \|y_e^{t+1}\|^2_{\Sigma_g}.
\]

Next, applying the elementary relations \(\langle u, v \rangle = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|u - v\|^2)\) to the first two terms on the left and \(\langle u, v \rangle_T = \frac{1}{2}(\|u\|^2_T + \|v\|^2 - \|u - v\|^2_T)\) to the third term on the left, multiplying both sides by a factor of 2 and rearranging terms, we arrive at

\[
\begin{align*}
\left( \frac{1}{\beta} \|x_e^{t+1}\|^2 + \|y_e^{t+1}\|^2_T \right) & - \left( \frac{1}{\beta} \|x_e^{t+1}\|^2 + \|y_e^{t+1}\|^2_T \right) \\
& \geq 2\|x_e^{t+1}\|^2_{\Sigma_f} + 2\|y_e^{t+1}\|^2_{\Sigma_g} + \|y^{t+1} - y^t\|^2_T + \beta \|By^{t+1}\|^2 - \beta \|Ax^{t+1}\|^2 \\
& = \|x_e^{t+1}\|^2_{2\Sigma_f - \beta A^* A} + 2\|y_e^{t+1}\|^2_{\Sigma_g} + \|y^{t+1} - y^t\|^2_T + \beta \|By^{t+1}\|^2 \geq 0.
\end{align*}
\]

(20)

Hence, the sequence \(\{\frac{1}{\beta} \|z^t_e\|^2 + \|y^t_e\|^2_T\}\) is monotonically nonincreasing (and thus bounded), and

\[
\lim_{t \to \infty} \|x^{t+1}_e\|^2_{2\Sigma_f - \beta A^* A} + 2\|y^{t+1}_e\|^2_{\Sigma_g} + \|y^{t+1} - y^t\|^2_T + \beta \|By^{t+1}\|^2 = 0.
\]

(21)

These together with the positive definiteness of \(\Sigma_g + T + \beta B^* B\) and \(2\Sigma_f - \beta A^* A\) imply that the sequence \(\{(x^t, y^t, z^t)\}\) is bounded. It then follows immediately that there exists a subsequence \(\{(x^t, y^t, z^t)\}\) that converges to a limit point \((x^\circ, y^\circ, z^\circ)\). We next show that \((x^\circ, y^\circ, z^\circ)\) satisfies \((17)\).

To see this, we first observe from \((21)\) and the positive definiteness of \(2\Sigma_f - \beta A^* A\) that

\[
\lim_{t \to \infty} \|x^{t+1}_e\| = 0, \quad \lim_{t \to \infty} \|y^t_e\|_{\Sigma_g} + \|By^t_e\| = 0, \quad \lim_{t \to \infty} \|y^{t+1} - y^t\|_T = 0.
\]

(22)
Furthermore, since
\[ \| Ax^e_{t+1} + By^e_{t+1} \| \leq \| A \| \| x^e_{t+1} \| + \| B y^e_{t+1} \|, \]
we must also have
\[ \lim_{t \to \infty} \| z^e_{t+1} - z^e \| = \beta \lim_{t \to \infty} \| Ax^e_{t+1} + By^e_{t+1} \| = 0. \tag{23} \]
Taking limits on both sides of (18) along the subsequence \( \{(x^{s-1}, y^{s-1}, z^{s-1})\} \), using (22), (23) and the closedness of the graphs of \( \partial f \) and \( \partial g \) [Page 80], we conclude that
\[ A^* z^\infty \in \partial f(x^\infty), \quad B^* z^\infty \in \partial g(y^\infty), \quad Ax^\infty + By^\infty - c = 0, \]
showing that \((x^\infty, y^\infty, z^\infty)\) satisfies (17).

To complete the proof, now it remains to show that \((x^\infty, y^\infty, z^\infty)\) is the unique limit of \(\{(x^t, y^t, z^t)\}\). Since \((x^\infty, y^\infty, z^\infty)\) satisfies (17), we could replace \((\bar{x}, \bar{y}, \bar{z})\) with \((x^\infty, y^\infty, z^\infty)\) in the above arguments, starting from (13). First, the convergence of \(\{x^t\}\) follows immediately from (22). Next, notice that the subsequence \(\{\frac{1}{\beta} \| z^t_e \|^2 + \| y^t_e \|^2_T\}\) converges to 0 as \(i \to \infty\). Since this sequence is also non-increasing, we must have
\[ \lim_{t \to \infty} \frac{1}{\beta} \| z^t_e \|^2 + \| y^t_e \|^2_T = 0. \tag{24} \]
From this, we see immediately that \(\lim_{t \to \infty} x^t = x^\infty\). Finally, using (24), (22) and the assumption that \(\Sigma_g + \mathcal{T} + \beta B^* B \succ 0\), we obtain further that \(\lim_{t \to \infty} y^t = y^\infty\). This completes the proof. \(\blacksquare\)

In the next theorem, we explore the iteration complexity of the above algorithm. We will make use of the Lagrangian function corresponding to (13), which is defined as follows:

\[ \ell(x, y, z) := f(x) + g(y) - \langle z, Ax + By - c \rangle. \tag{25} \]

Recall from [30] Theorem 28.4 that for any \((\bar{x}, \bar{y}, \bar{z})\) satisfying (17), one has
\[ \ell(x, y, z) \geq \ell(\bar{x}, \bar{y}, \bar{z}) = f(\bar{x}) + g(\bar{y}) = \ell(\bar{x}, \bar{y}, \bar{z}), \quad \forall (x, y, z). \tag{26} \]
Thus, for any sequence \(\{(x^t, y^t, z^t)\}\) converging to a point \((\bar{x}, \bar{y}, \bar{z})\) satisfying (17), the quantity \(E(x^t, y^t) := \ell(x^t, y^t, z^t) - \ell(\bar{x}, \bar{y}, \bar{z})\) is always nonnegative. Furthermore, in view of (26), it holds that \((\bar{x}, \bar{y})\) is a solution of (13) if and only if
\[ E(\bar{x}, \bar{y}) = 0 = \| A\bar{x} + B\bar{y} - c \|. \]
In the next theorem, we study the iteration complexity of the proximal AMA by establishing upper bounds on \(E(x, y)\) and \(\|Ax + By - c\|\) along a suitable sequence. Our proof is similar to [21] Theorem 3.5] which established a bound similar to (27) for their algorithm.

**Theorem 2.** Suppose that A1 holds and let \(\{(x^t, y^t, z^t)\}\) be generated from the proximal AMA. Suppose further that \(\Sigma_g + \mathcal{T} + \beta B^* B \succ 0\) and \(2\Sigma_f - \beta A^* A \succeq \delta I\) for some \(\delta > 0\). Define
\[ (\bar{x}^N, \bar{y}^N) = \frac{1}{N} \sum_{i=1}^{N} (x^t, y^t). \]
Then it holds that
\[ \ell(\bar{x}^N, \bar{y}^N, \bar{z}) - \ell(\bar{x}, \bar{y}, \bar{z}) \leq \frac{\delta + \beta \| A \|^2}{2\delta N} \left( \frac{1}{\beta} \| z^0 - \bar{z} \|^2 + \| y^0 - \bar{y} \|^2_T \right), \tag{27} \]
and
\[ \| A\bar{x}^N + B\bar{y}^N - c \| \leq \frac{1}{\sqrt{N}} \left( \frac{\| A \|}{\sqrt{\delta}} + \frac{1}{\sqrt{\beta}} \right) \sqrt{\frac{1}{\beta} \| z^0 - \bar{z} \|^2 + \| y^0 - \bar{y} \|^2_T}, \tag{28} \]
where \((\bar{x}, \bar{y}, \bar{z})\) is the limit of \(\{(x^t, y^t, z^t)\}\) guaranteed by Theorem [7].
Proof. We first establish \textbf{(27)}. Let \((\bar{x}, \bar{y}, \bar{z})\) be the limit of \(\{(x^t, y^t, z^t)\}\), and write \(w^t := w^t - \bar{w}\), where \(w\) represents \(z, y\) and \(z\) respectively, as in the proof of Theorem \textbf{11} for notational simplicity. Then it holds that

\[
\langle A^*(z^{t+1} - z^t), x^{t+1} \rangle + \langle T(y^{t+1} - y^t), y^{t+1} \rangle + \frac{1}{\beta}(z^{t+1} - z^t, z_e^{t+1})
\]

\[
= \frac{1}{\beta}(z^{t+1} - z^t, \beta A x_e^{t+1}) + \langle y^{t+1} - y^t, y^{t+1} \rangle_T + \frac{1}{\beta}(z^{t+1} - z^t, z_e^{t+1})
\]

\[
= \frac{1}{2\beta} \|z^{t+1} - z^t + \beta A x_e^{t+1}\|^2 - \frac{1}{2\beta} \|z^{t+1} - z^t\|^2 - \frac{1}{2\beta} \|A x_e^{t+1}\|^2
\]

\[
+ \frac{1}{\beta} \|y^{t+1} - y^t\|_T^2 + \frac{1}{\beta} \|y^{t+1}\|_T^2 - \frac{1}{2}\|y^t\|_T + \frac{1}{\beta} \|z^{t+1} - z^t\|^2 + \frac{1}{2\beta} \|z_e^{t+1}\|^2 - \frac{1}{2\beta} \|z_e^t\|^2
\]

\[
= \frac{\beta}{2} \|B y_e^{t+1}\|^2 - \frac{\beta}{2} \|A x_e^{t+1}\|^2 + \frac{1}{\beta} \|y^{t+1} - y^t\|_T^2 + \frac{1}{\beta} \|y^{t+1}\|_T^2 - \frac{1}{2}\|y^t\|_T + \frac{1}{\beta} \|z^{t+1}\|_T^2 - \frac{1}{\beta} \|z^t\|_T^2
\]

\[
\geq - \frac{\beta}{2} \|A x_e^{t+1}\|^2 + \frac{1}{\beta} \|y^{t+1}\|_T^2 - \frac{1}{\beta} \|y^t\|_T + \frac{1}{\beta} \|z^{t+1}\|_T^2 - \frac{1}{\beta} \|z^t\|_T^2
\]

(29)

where we made use of the elementary relations \((u, v) = \frac{1}{2}(||u||^2 + ||v||^2 - ||u - v||^2) = \frac{1}{2}(||u||^2 + ||v||^2 - ||u - v||^2)\) and \((u, v)_T = \frac{1}{2}(||u||_T^2 + ||v||_T^2 - ||u - v||_T^2)\) in the second equality, and the fact that \(-A\bar{x} = B\bar{y} - c\) and the definition of \(z^{t+1}\) in the third equality. Moreover, notice from \textbf{(13)} that the iterates of the proximal AMA satisfy the following relations:

\[
\begin{cases}
- A^*(z^{t+1} - z^t) \in \partial f(x^{t+1}) - A^* z^{t+1}, \\
- T(y^{t+1} - y^t) \in \partial g(y^{t+1}) - B^* z^{t+1}, \\
- \frac{1}{\beta}(z^{t+1} - z^t) = A x^{t+1} + B y^{t+1} - c.
\end{cases}
\]

Using this observation, \textbf{(29)}, and the fact that the Lagrangian function \textbf{(25)} is convex in \((x, y)\) and linear in \(z\), we obtain that

\[
- \frac{\beta}{2} \|A x_e^{t+1}\|^2 + \frac{1}{\beta} \|y^{t+1}\|_T^2 - \frac{1}{\beta} \|y^t\|_T + \frac{1}{\beta} \|z^{t+1}\|_T^2 - \frac{1}{\beta} \|z^t\|_T^2
\]

\[
\leq - \langle A^*(z^{t+1} - z^t), \bar{x} - x^{t+1} \rangle + \langle -T(y^{t+1} - y^t), \bar{y} - y^{t+1} \rangle - \frac{1}{\beta}(z^{t+1} - z^t, \bar{z} - z^{t+1})
\]

\[
\leq \ell(\bar{x}, \bar{y}, \bar{z}^{t+1}) - \ell(x^{t+1}, y^{t+1}, z^{t+1}) - \frac{1}{\beta}(z^{t+1} - z^t, \bar{z} - z^{t+1})
\]

\[
= \ell(\bar{x}, \bar{y}, \bar{z}^{t+1}) - \ell(x^{t+1}, y^{t+1}, \bar{z}) = \ell(\bar{x}, \bar{y}, \bar{z}) - \ell(x^{t+1}, y^{t+1}, \bar{z}).
\]

Rearranging terms in the above inequality and summing from \(t = 0\) to \(N - 1\), we obtain that

\[
\sum_{t=0}^{N-1} \frac{1}{2}(\|z^t\|^2 + ||y^t||_T^2) + \frac{\beta}{2} \sum_{t=0}^{N-1} \|A x_e^{t+1}\|^2.
\]

(30)

We next derive an upper bound for \(\sum_{t=0}^{N-1} \|A x_e^{t+1}\|^2\). Since \(2\Sigma f - \beta A^* A \succeq \delta I\), we observe from \textbf{(20)} that

\[
\|x_e^{t+1}\|^2 \leq \frac{1}{\delta} \|x_e^{t+1}\|^2 + \frac{1}{\delta} \|y^t\|_T^2 - \frac{1}{\delta} \|z^t\|^2 + ||y^t||_T^2.
\]

Hence, we obtain that

\[
\sum_{t=0}^{N-1} \|A x_e^{t+1}\|^2 \leq \|A\|^2 \sum_{t=0}^{N-1} \|x_e^{t+1}\|^2 \leq \frac{\|A\|^2}{\delta} \left( \frac{1}{\beta} \|z^0\|^2 + \|y_0\|_T^2 - \frac{1}{\beta} \|z_e^0\|^2 - ||y_0||_T^2 \right).
\]

(31)
Combining this bound with \( \text{[32]} \), we have that
\[
\sum_{t=0}^{N-1} \left( \ell(x^{t+1}, y^{t+1}, z) - \ell(\bar{x}, \bar{y}, \bar{z}) \right) \leq \left( \frac{1}{2} + \frac{\beta}{\delta} \right) \left( \frac{1}{\beta} \|z^0\|^2 + \|y^0\|^2_\gamma \right).
\]

The bound \( \text{[27]} \) now immediately follows from this and the convexity of \( \ell \) in the \((x, y)\) variables.

We next prove \( \text{[28]} \). Summing \( \text{[31]} \) from \( t = 0 \) to \( N - 1 \) and using the definition of \( \bar{x}^N \), we have
\[
\|\bar{x}^N - \bar{x}\|^2 \leq \frac{1}{N} \sum_{t=0}^{N-1} \|x^{t+1}\|^2 \leq \frac{1}{N\delta} \left( \frac{1}{\beta} \|z^0\|^2 + \|y^0\|^2_\gamma \right).
\]

Also, we observe from \( \text{[20]} \) that
\[
\|B\beta y^{t+1}\|^2 \leq \frac{1}{\beta} \left( \frac{1}{\beta} \|z^0\|^2 + \|y^0\|^2_\gamma \right) - \frac{1}{\beta} \left( \frac{1}{\beta} \|z^{t+1}\|^2 + \|y^{t+1}\|^2_\gamma \right).
\]

Summing this relation from \( t = 0 \) to \( N - 1 \) and using the definition of \( \bar{y}^N \), we have further that
\[
\|B\bar{y}^N - \bar{y}\|^2 \leq \frac{1}{N} \sum_{t=0}^{N-1} \|B\beta y^{t+1}\|^2 \leq \frac{1}{N\beta} \left( \frac{1}{\beta} \|z^0\|^2 + \|y^0\|^2_\gamma \right).
\]

Using \( \text{[32]} \), \( \text{[33]} \) and the triangle inequality, one obtains \( \text{[28]} \). This completes the proof. \( \square \)

The proximal AMA is closely related to the proximal ADMM, which has been studied extensively recently in the literature; see, for example, [5, 9, 13, 38, 39, 41]. In essence, the latter algorithm is also applicable to solve \( \text{[13]} \) and is obtained by replacing the first subproblem in \( \text{[16]} \) with
\[
x^{t+1} = \arg\min_x \left\{ f(x) - \langle z^t, Ax \rangle + \frac{\beta}{2} \|Ax + By^t - c\|^2 + \frac{1}{2} \|x - x^t\|^2_\mathcal{S} \right\},
\]
for some positive semidefinite linear map \( \mathcal{S} \). In [9, Theorem B.1], this algorithm is shown to converge under conditions similar to those in Theorem [1] but has no restrictions in \( \beta > 0 \), and allows an extra freedom in picking the stepsize in the \( z \)-update to be \( \gamma \beta \) instead of just \( \beta \), as long as \( \gamma \in (0, \frac{\sqrt{\beta} + 1}{2\beta}) \).

In the case when the proximal mapping of \( \frac{1}{\beta} f \) is easy to compute, the subproblem \( \text{[34]} \) for the proximal ADMM could be easy to solve with a suitable choice of \( \mathcal{S} \). Due to the extra freedom in choosing parameters, it is conceivable that the proximal ADMM will work better than the proximal AMA. As a consequence, for AMA, we do not consider the case when the proximal mapping of \( \frac{1}{\beta} f \) is easy to compute and thus did not add proximal term to the first subproblem of the proximal AMA.

### 4 The proximal-proximal gradient algorithm

In this section, we apply the proximal AMA \( \text{[16]} \) developed in Section \( \text{[3]} \) to \( \text{[9]} \), or equivalently, \( \text{[14]} \). To this end, letting \( \tau \geq \beta \|\mathcal{M}^*\| \) so that \( \mathcal{T} := \tau \mathcal{I} - \beta \mathcal{M} \mathcal{M}^* \succeq 0 \), we replace the second subproblem in \( \text{[16]} \) by
\[
y^{t+1} = \arg\min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \|x^{t+1} + \mathcal{M}^* y\|^2 + \frac{1}{2} \|y - y^t\|^2_\tau \right\},
\]
where \( \| \cdot \|_\tau \) is the semi-norm induced by the positive semidefinite linear map \( \mathcal{T} \). Notice that the optimization problem for this new subproblem has a unique solution due to the choice of \( \mathcal{T} \). In
particular, the iterates for this new algorithm are all well-defined, i.e., all the subproblems that arise have (unique) minimizers.

We next discuss how the subproblems of this new algorithm as applied to (11) (or, equivalently, the proximal AMA as applied to (10)) can be solved efficiently. From the first-order optimality condition, the first subproblem in (11) amounts to finding \( x^{t+1} \) so that \( \nabla h(z^t) = 0 \), which can be obtained by setting \( x^{t+1} = \nabla h(z^t) \) by [30, Theorem 23.5]. Moreover, the second subproblem (35) can be reformulated as

\[
y^{t+1} = \arg \min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, M^*y \rangle + \frac{\beta}{2} \|x^{t+1} + M^*y\|^2 + \frac{1}{2} \|y - y^t\|^2 \right\}
\]

\[
= \arg \min_y \left\{ P^*(y) + \langle b - Mz^t + \beta Mx^{t+1} + \beta MM^*y^t, y \rangle + \frac{\tau}{2} \|y - y^t\|^2 \right\}
\]

\[
= \arg \min_y \left\{ P^*(y) + \frac{\tau}{2} \|y - (\frac{b - Mz^t + \beta Mx^{t+1} + \beta MM^*y^t}{\tau})\|^2 \right\}
\]

\[
= \text{prox}_{\tau^{-1}P^*} \left( \frac{Ty^t - b + Mz^t - \beta Mx^{t+1}}{\tau} \right),
\]

where \( T = \tau I - \beta MM^* \). Since the proximal mapping of \( \tau P \) is easy to compute, the same is true of \( \tau^{-1}P^* \), due to [8]. Precisely, we have

\[
y^{t+1} = \frac{Ty^t - b + Mz^t - \beta Mx^{t+1}}{\tau} - \arg \min_u \left\{ P(\tau u) + \frac{\tau}{2} \left\| u - \frac{Ty^t - b + Mz^t - \beta Mx^{t+1}}{\tau} \right\|^2 \right\}
\]

\[
= \tau^{-1} \left( Ty^t - b + Mz^t - \beta Mx^{t+1} \right) - \arg \min_v \left\{ P(v) + \frac{1}{2\tau} \left\| v - (Ty^t - b + Mz^t - \beta Mx^{t+1}) \right\|^2 \right\}
\]

\[
= \tau^{-1} \left( Ty^t - b + Mz^t - \beta Mx^{t+1} - \text{prox}_{\tau P}(Ty^t - b + Mz^t - \beta Mx^{t+1}) \right)
\]

Thus, the second subproblem (35) can also be efficiently solved.

We note that the only difference between this new algorithm and the proximal gradient algorithm (11) lies in the second subproblem in (11) (and also the choice of stepsize \( \beta > 0 \)), where we simplified the subproblem by adding a proximal term. With our choice of \( T = \tau I - MM^* \), the subproblem is reduced to a computation of the proximal mapping of \( \frac{1}{\tau}P^* \). Hence, we shall also refer to this new algorithm as the proximal-proximal gradient (PPG) algorithm.

We now summarize the PPG algorithm for solving (1) as follows:

**PPG algorithm for (1)**

**Step 0.** Input \((y^0, z^0) \in \mathcal{Y} \times \mathcal{Z}, \beta \in (0, \frac{1}{\tau}), \tau \geq \beta \|M^*M\| \) and \( T = \tau I - MM^* \).

**Step 1.** Set

\[
\begin{align*}
y^{t+1} &= \text{prox}_{\tau^{-1}P^*} \left( \frac{Ty^t - b + Mz^t - \beta Mx^{t+1}}{\tau} \right), \\
z^{t+1} &= z^t - \beta(\nabla h(z^t) + M^*y^{t+1}).
\end{align*}
\]

**Step 2.** If a termination criterion is not met, go to Step 1.

We have the following convergence result concerning this new algorithm, which is a consequence of Theorem [1]

**Theorem 3.** Let \( \{(y^t, z^t)\} \) be generated from the PPG algorithm for solving (11) and define \( x^{t+1} = \nabla h(z^t) \) for all \( t \geq 0 \). Then \((x^t, y^t)\) converges to an optimal solution to (9) and \((z^t)\) converges to an optimal solution to (11).
Proof. Recall that \( \{(x^t, y^t, z^t)\} \) thus generated is the same as the sequence obtained from the proximal AMA for solving [5], using the same initial points, \( \beta \) and \( T \).

We first show that \( A_1 \) holds for [9]. To this end, let \( \bar{z} \) be a solution of [11]. Since \( h \) is continuous and differentiable, from the first-order optimality condition, [2] and the subdifferential calculus rules [30, Theorem 23.8], [30, Theorem 23.9], we have

\[
0 \in \nabla h(\bar{z}) + \partial P(\bar{M} \bar{z} - b).
\]

Hence, there exists \( \bar{y} \in \partial P(\bar{M} \bar{z} - b) \) and \( \bar{x} := \nabla h(\bar{z}) \) so that \( \bar{x} + \partial \bar{y} = 0 \). Furthermore, notice from [30, Theorem 23.5] that

\[
\bar{y} \in \partial P(\bar{M} \bar{z} - b) \Rightarrow \bar{M} \bar{z} - b \in \partial P(\bar{y}) \quad \text{and} \quad \bar{x} = \nabla h(\bar{z}) \Rightarrow \bar{z} \in \partial h^*(\bar{x}).
\]

We thus conclude that [17] is satisfied with \( h^*(x) \) in place of \( f(x) \), \( P^*(y) + \langle h, y \rangle \) in place of \( g(y) \), the identity map \( I \) in place of \( A \), \( M^* \) in place of \( B \) and \( 0 \) in place of \( c \).

Next, recall from [25, Theorem 2.1.5] that for any \( z_1, z_2 \in Z \), we have

\[
\langle \nabla h(z_1) - \nabla h(z_2), z_1 - z_2 \rangle \geq \frac{1}{L} \|z_1 - z_2\|^2,
\]

since \( \nabla h \) is Lipschitz continuous with modulus bounded by \( L \). Hence \( h^* \) satisfies [14] in place of \( f \) with \( \Sigma_f = \frac{1}{L} I \) and \( \Sigma_g = 0 \).

The conclusion of the theorem now follows immediately from a direct application of Theorem [11] with \( \Sigma_f = \frac{1}{L} I \) and \( \Sigma_g = 0 \). \( \square \)

Remark 1. Unlike the proximal gradient algorithm [3], the sequence \( \{Mz^t - b\} \) generated using \( \{z^t\} \) from the above algorithm does not necessarily lie in dom \( P \). This could be a disadvantage since then the objective function \( F \) of [11] is not necessarily defined at each \( z^t \). However, notice from [35] that

\[
0 \in \partial P^*(y^{t+1}) + b - Mz^t + \beta M(x^{t+1} + M^* y^{t+1}) + T(y^{t+1} - y^t)
\]

\[
\Rightarrow y^{t+1} \in \partial P(Mz^{t+1} - b - T(y^{t+1} - y^t)).
\]

Hence, in the case when \( M \) is surjective so that \( T = MR \) for some linear map \( R \), one can define

\[
\tilde{z}^{t+1} := z^{t+1} - R(y^{t+1} - y^t).
\]

Then we see immediately from [37] that \( \{\tilde{z}^t\} \subseteq \text{dom} \ P \). Furthermore, since \( \{y^t\} \) is convergent by Theorem [3] it follows that \( \lim_{t \to \infty} \tilde{z}^t = \lim_{t \to \infty} z^t \). Thus, the sequence \( \{z^t\} \) is a feasible minimizing sequence for [11].

The PPG algorithm takes a particularly simple form when \( h(z) = \frac{1}{L} \|z - \bar{z}\|^2 \) for some \( \bar{z} \in Z \), as is illustrated in the next example.

Example 1. Fix any \( \bar{z} \in Z \) and consider the following optimization problem:

\[
\min_z \frac{1}{L} \|z - \bar{z}\|^2 + P(Mz - b),
\]

which is a special case of [11] with \( h(z) = \frac{1}{L} \|z - \bar{z}\|^2 \). Since \( \nabla h(z) = z - \bar{z} \), it is easy to see that one can take \( L = 1 \). Now, taking \( \beta = 1, \tau \geq \|M^* M\| \) and \( T = \tau I - MM^* \) in the PPG algorithm, it is routine to show that

\[
\begin{align*}
y^{t+1} &= \text{prox}_{\tau, -1 P^*} \left( (\tau I - MM^*) y^t + M \bar{z} - b \right), \\
z^{t+1} &= \bar{z} - M^* y^{t+1},
\end{align*}
\]

(39)
notice that the above updating rules are independent of $z^t$. In particular, one can completely ignore the $z$-update in the course of the algorithm. Furthermore, if $M$ is surjective with $\mathcal{R} = \mathcal{MR}$ for some linear map $\mathcal{R}$, one can obtain a feasible sequence $\{\tilde{z}^t\}$ converging to the optimal solution as

$$z^{t+1} = \tilde{z} - M^* y^{t+1} - \mathcal{R}(y^{t+1} - y^t).$$

On the other hand, one can show that the $y$-update in (39) is exactly the updating rule obtained by applying the proximal gradient algorithm (13) to the dual of (35). Indeed, according to (10), the dual of (35) is given by

$$\min_y \left[ \frac{1}{2} ||M^* y||^2 - \langle M\tilde{z}, y \rangle + P^*(y) \right].$$

Notice that $\tau$ is an upper bound of the Lipschitz continuity modulus of the gradient of the smooth part $h_2(y)$ of the objective function in (40). Thus, the updating rule (13) applied to (40) yields the iterates

$$y^{t+1} = \text{prox}_{\tau^{-1}P^*}\left(y^t - \frac{1}{\tau}(M^*y^t - M\tilde{z} + b)\right),$$

which is just the $y$-update in (39).

We illustrate the PPG algorithm when $M^*M = I$ or $MM^* = I$ in the next example.

**Example 2.** Suppose that $M^*M = I$ so that $||M^*|| = 1$. One can then take $\tau = \beta$ in the PPG algorithm. Using the definition of $z^{t+1}$, (30) and introducing the auxiliary iterates $u^t$ and $w^t$, one can easily verify that the iterates of the algorithm can be represented as

$$\begin{cases} w^{t+1} = Ty^t - b + Mz^t - \beta M\nabla h(z^t), & u^{t+1} = \text{prox}_{\tau^{-1}P}(w^{t+1}), \\
 z^{t+1} = M^* (u^{t+1} + b), & y^{t+1} = \frac{1}{\tau}(w^{t+1} - u^{t+1}). \end{cases}$$

On the other hand, if $MM^* = I$, then $T = 0$. Hence, no proximal term is added to the alternating minimization algorithm when applied to the dual problem (9), and the PPG algorithm for (11) reduces to the proximal gradient algorithm (3).

We consider the sum of several functions with simple proximal mappings in the next example.

**Example 3.** Consider the following optimization problem:

$$\min_z h(z) + \sum_{i=1}^m P_i(z), \quad (41)$$

where the proximal mappings for $\tau P_i$, $\tau > 0$ and $i = 1, ..., m$ are all easy to compute. The problem (11) is readily written in the form of (11) and hence the PPG algorithm is applicable. More precisely, define $Mz = (z, z, ..., z) \in Z^m$ and $P(u_1, ..., u_m) = \sum_{i=1}^m P(u_i)$. Then we have $\sum_{i=1}^m P_i(z) = P(Mz)$. Moreover, the proximal mapping of $\tau P$, $\tau > 0$, is easy to compute.

Thanks to Theorem 2 one can obtain various complexity results for the PPG algorithm by using different choices of parameters. Here is one such result.

**Corollary 1.** Let $\{(y^t, z^t)\}$ be generated from the PPG algorithm for solving (11) with $\beta = \frac{1}{\tau}$ and $\tau = \beta \sigma(M^*M)$. Define $x^{t+1} = \nabla h(z^t)$ for all $t \geq 0$ and

$$(\bar{x}^N, \bar{y}^N) = \frac{1}{N} \sum_{t=1}^N (x^t, y^t).$$
Then it holds that

\[
\ell(\bar{x}^N, \hat{y}^N, z) - \ell(\bar{x}, \bar{y}, z) \leq \frac{1}{N} \left( Lz^0 - \bar{z} \right)^2 + \|y^0 - \bar{y}\|_{I - \beta M^*M}^2,
\]

and

\[
\|\bar{x}^N + M^*\hat{y}^N\| \leq 2 \sqrt{\frac{L}{N}} \sqrt{Lz^0 - \bar{z}}^2 + \|y^0 - \bar{y}\|_{I - \beta M^*M}^2,
\]

where \((\bar{x}, \bar{y}, \bar{z})\) is the limit of \:\{(x^t, y^t, z^t)\}\: guaranteed by Theorem 3 and

\[
\ell(x, y, z) = h^*(x) + P^*(y) + \langle b, y \rangle - \langle z, x + M^*y \rangle.
\]

**Proof.** Recall that the PPG algorithm in this case is just the proximal AMA applied to (9) with \(y^0\) solution of (42) from an optimal solution \(\tilde{x}\) can equivalently solve for (44). Moreover, it follows from (43) that one can recover the optimal solution of (42) if each subproblem (3) and (7). Furthermore, from (2) and [30, Theorem 31.2], we see that (42) indeed equals (44). This implies that equality holds throughout in the above relations. Thus, in order to solve for (42), one can equivalently solve for (44). Moreover, it follows from (43) that one can recover the optimal solution of (42) from an optimal solution \(\hat{y}^t\) of (44) by setting

\[
\hat{y}^{t+1} = \hat{y}^t + \frac{1}{L} \nabla h(z^t) + M^*\hat{y}^t.
\]

We thus discuss an algorithm for solving (44) instead. Recall that the proximal mapping of \(\tau^{-1}P^*\) is easy to compute for \(\tau > 0\), assuming that of \(\tau P, \tau > 0\), is easy to compute; see [8] and the discussions leading to [30]. Hence, the maximization problem in (44) can be efficiently solved using the proximal gradient algorithm. In an inexact proximal gradient algorithm based on
this scheme, one solves (44) sufficiently accurately by the proximal gradient algorithm and updates $z^{t+1}$ according to (45), using the approximate solution in place of $\tilde{y}^{t+1}$.

However, instead of getting a sufficiently accurate solution for (44), suppose we go to the other extreme: apply one step of the proximal gradient algorithm to (44), initialized at $y^t$. Then, the resulting point $y^{t+1}$ is given by

$$y^{t+1} = \text{prox}_{\tau^{-1}p^*} \left( y^t - \frac{\beta M^*M y^t - M z^t + \beta M \nabla h(z^t) + b}{\tau} \right)$$

(46)

where $\beta = \frac{1}{L}$ and $\tau \geq \beta \|M^*M\|$, the Lipschitz continuity modulus of the gradient of $h_3$ in (44). Notice that the PPG algorithm just consists of (46) and (45), with $\tilde{y}^{t+1}$ replaced by $y^{t+1}$. In this sense, the PPG algorithm can be viewed as an inexact proximal gradient algorithm with an (very) inaccurately solved subproblem, where we only take one step of the proximal gradient algorithm to solve the dual of the subproblem. However, this approach is fundamentally different from those considered in [32] which require the subproblem to be solved up to a certain accuracy. Hence, their convergence results do not directly apply to our algorithm. On the other hand, the result in [28, Section 4.2] is closer in spirit to our approach; However, unlike there, in our algorithm, $z^{t+1} - z^t$ is not a descent direction for $F(z)$ in general and no line search is needed in our algorithm. Hence, their convergence results also do not directly apply to our algorithm.

5 Numerical results

In this section, we perform numerical experiments to illustrate the performance of our algorithm. We consider two applications: the system realization problem modeled via nuclear norm minimization [18, Section II.B] and the fused lasso logistic regression problem [21, Section 5]. Our codes are written in MATLAB. All numerical experiments are performed on an SGI XE340 system, with two 2.4 GHz quad-core Intel E5620 Xeon 64-bit CPUs and 48 GB RAM, equipped with SUSE Linux Enterprise server 11 SP1 and MATLAB 7.14 (R2012a). All routines are timed using the tic-toc function in MATLAB.

5.1 System realization problem

As was formulated in [18, Section II.B] (see also [9, Section 5]), the system realization problem modeled as a nuclear norm minimization problem takes the following form:

$$\min_{\mathbf{z} \in \mathbb{R}^{m \times n(j+k-1)}} \mathcal{P}_{\text{sys}}(\mathbf{z}) = \frac{1}{2} \| w \circ \mathbf{z} - w \circ \tilde{z} \|^2 + \lambda \| \mathcal{H}(\mathbf{z}) \|_*,$$

(47)

where $\mathbf{z} = (z_0 \ z_1 \ \cdots \ z_{j+k-2})$ with each $z_i \in \mathbb{R}^{m \times n}$, $\tilde{z}$ is the given measurement, $w = (w_0 \ w_1 \ \cdots \ w_{j+k-2}) \in \mathbb{R}^{m \times n(j+k-1)}$ is a zero-one matrix that are ones for the blocks with $i = 0, ..., k - 1$ and is zero otherwise, $\circ$ denotes the Hadamard (entry-wise) product, $\| \cdot \|$ denotes the Frobenius norm (the norm induced by the trace inner product on $\mathbb{R}^{m \times n}$), $\lambda > 0$ is the regularization parameter, $\| \cdot \|_*$ denotes the nuclear norm (the sum of all singular values), and $\mathcal{H}(\mathbf{z})$ is a block Hankel matrix defined as

$$\mathcal{H}(\mathbf{z}) = \begin{pmatrix} z_0 & z_1 & \cdots & z_k \\ z_1 & z_2 & \cdots & z_k \\ \vdots & \vdots & \ddots & \vdots \\ z_{j-1} & z_j & \cdots & z_{j+k-2} \end{pmatrix} \in \mathbb{R}^{mj \times nk}.$$
It is easy to see that (17) is in the form of (1) with
\[ h(z) = \frac{1}{2} \| w \circ z - w \circ \hat{z} \|^2, \quad P(u) = \lambda \| u \|_*, \quad M = \mathcal{M}, \quad b = 0. \]

From these we see that the condition (2) is trivially satisfied. Moreover, as mentioned in the introduction, the proximal mapping of \( \tau P, \tau > 0 \), is easy to compute. Furthermore, since \( z \mapsto \|\mathcal{H}(z)\| \) is coercive, the set of optimal solutions of (17) is nonempty. Hence, all three assumptions on (1) are satisfied for the specific problem (17). For ease of reference, we also write down the dual of (17) as follows:
\[
\min_{\nu, y} \quad d_{\text{sys}}(\nu) := \frac{1}{2} \| \nu \|^2 + \langle w \circ \hat{z}, \nu \rangle \\
\text{s.t.} \quad w \circ \nu + \mathcal{H}^*(y) = 0, \quad \|y\|_{\infty} \leq \lambda, \tag{48}
\]
where \( \|y\|_{\infty} \) denotes the operator norm of \( y \in \mathbb{R}^{m \times n} \), where \( \nu \) is related to the \( x \) in (9) via \( x = w \circ \nu \).

In this subsection, we will perform numerical experiments to compare our PPG algorithm with the MFBS method (6) and the primal ADMM2 in (9) Section 5.1 for solving (17). To apply our algorithm, we have to determine a bound \( L \) for the Lipschitz continuity modulus of \( \nabla h \), and obtain a bound of \( \|\mathcal{M}'\mathcal{M}\| \). It is easy to see one can take \( L = 1 \). Furthermore, it follows from (9) Section 2 that \( \|\mathcal{H}'\mathcal{H}\| \leq \min\{j, k\} \). On the other hand, to apply the MFBS method (6), we need a bound \( L_{\mathcal{M}} \) of the Lipschitz continuity modulus of \( G(z, y) \). By (23) Section 6.2, \( L_{\mathcal{M}} \) can be chosen to be \( \frac{1}{L}(1 + \sqrt{1 + 4 \min\{j, k\}}) \). Finally, we refer the readers to (9) Section 5.1 for details about the primal ADMM.

We initialize all three algorithms at the origin, i.e., \( (y^0, z^0) = (0, 0) \). We terminate the algorithms when
\[
\max \left\{ \frac{\min_{s \in \Gamma} p_{\text{sys}}(z^*) + d_{\text{sys}}(-w \circ \mathcal{H}^* (\text{proj}_\Omega(y^t)))}{\max\{\min_{s \in \Gamma} p_{\text{sys}}(z^*), 1\}}, \frac{5\|\mathcal{H}^* (\text{proj}_\Omega(y^t)) - w \circ \mathcal{H}^* (\text{proj}_\Omega(y^t))\|}{\max\{\|\mathcal{H}^* (\text{proj}_\Omega(y^t))\|, 1\}} \right\} < \text{tol} \tag{49}
\]
for some \( \text{tol} > 0 \), where \( \text{proj}_\Omega \) denotes the projection onto the set \( \Omega := \{ y : \|y\|_{\infty} \leq \lambda \} \) the set \( \Gamma := \{ s : 1 \leq s \leq t, s \text{ is a multiple of } 10 \} \), and the above criterion (52) is checked every 10 iterations. In our test below, we take \( \text{tol} = 1e - 4 \) for all three algorithms. Also, for our PPG algorithm, we take \( \beta = 1 = \frac{1}{L} \) for \( \lambda = 0.05 \) and \( \beta = 0.05 \) otherwise, \( \tau = \beta \min\{j, k\} \), while for the MFBS method (6), we take \( L_{\mathcal{M}} = \frac{1}{L}(1 + \sqrt{1 + 4 \min\{j, k\}}) \) and \( \sigma = 0.95 \). We use the same parameters as used in (9) Section 5.1 for the primal ADMM.

We generate random instances as in (18) Section II(B)]. We start by generating random matrices \( A \in \mathbb{R}^{r \times r}, B \in \mathbb{R}^{r \times n} \) and \( C \in \mathbb{R}^{s \times r} \) with i.i.d. standard Gaussian entries and normalize them to have operator norm 1. We then randomly generate a \( v_0 \sim N(0, I) \) and \( e_t \sim N(0, I) \) for \( t = 0, ..., T - 1 \) with i.i.d. standard Gaussian entries, and create an “output” \( \hat{u}_t, t = 0, ..., T - 1 \), according to the state space model:
\[
v_{t+1} = Av_t + Be_t, \\
\hat{u}_t = Cv_t + e_t.
\]
Random noise is then added to the output \( \hat{u} \) to give \( \hat{u} = \hat{u} + \sigma \epsilon \), with \( \epsilon \) having i.i.d. standard Gaussian entries. Finally, we set, for each \( i = 0, ..., k - 1 \),
\[
\hat{z}_i = \frac{1}{T} \sum_{t=0}^{T-1-i} \hat{u}_{t+i}^T, \tag{52}
\]

\footnote{For the MFBS method, we used \((u^t, v^t)\) in place of \((z^t, y^t)\) in (19). On the other hand, the projection in (19) is only performed for the primal ADMM.}

\footnote{We experimented with \( \beta = 0.05, \frac{1}{L} \) and \( \frac{1.95}{L} \). Setting \( \beta = 0.05 \) seems to work best for \( \lambda \geq 0.1 \); while for \( \lambda = 0.05 \), setting \( \beta = \frac{1}{L} \) works best.}
and $\hat{z}_i = 0$ for $i \geq k$.

In the test below, we fix $T = 1000$, $m = n = 10$, $r = 10$, $j = 21$ and $\sigma = 5 \times 10^{-2}$. For each $k = 100, 200$ and $300$, and $\lambda = 0.05, 0.1$ and $0.5$, we generate 10 random instances as described above. The computational results, averaged over the 10 instances, are reported in Table 1 where we report the number of iterations (iter), CPU time in seconds (cpu), primal objective value (pobj) at termination, dual objective value (dobj) at termination and the relative dual infeasibility (dfeas).

\[
\frac{\|\mathcal{H}^*(\text{proj}_I(y^f)) - w \circ \mathcal{H}^*(\text{proj}_I(y^f))\|}{\max(\|\mathcal{H}^*(\text{proj}_I(y^f))\|, 1)}.
\]

We observe that the primal ADMM is the fastest algorithm, and our PPG algorithm is usually faster than the MFBS method.

| $k$  | $\lambda$ | PPG               | P. ADMM2          | MFBS             |
|------|----------|-------------------|-------------------|------------------|
|      |          | iter  | cpu   | pobj/dobj/dfeas | iter  | cpu   | pobj/dobj/dfeas | iter  | cpu   | pobj/dobj/dfeas |
| 100  | 0.05     | 123   | 7.0   | 6.073e+0/6.072e+0/5.7e-6 | 40    | 2.5   | 6.073e+0/6.073e+0/9.6e-6 | 108   | 6.7   | 6.073e+0/6.073e+0/1.5e-5 |
| 100  | 0.10     | 105   | 6.2   | 7.419e+0/7.419e+0/1.4e-5 | 78    | 5.0   | 7.419e+0/7.419e+0/4.2e-7 | 299   | 19.6  | 7.419e+0/7.419e+0/1.9e-6 |
| 100  | 0.50     | 80    | 4.5   | 1.180e+1/1.180e+1/2.6e-6 | 21    | 1.3   | 1.180e+1/1.180e+1/4.2e-8 | 97    | 6.0   | 1.180e+1/1.180e+1/4.8e-6 |
| 200  | 0.05     | 44    | 5.0   | 1.014e+1/1.014e+1/1.5e-5 | 40    | 4.9   | 1.014e+1/1.014e+1/5.2e-6 | 191   | 23.3  | 1.014e+1/1.014e+1/9.1e-6 |
| 200  | 0.10     | 146   | 19.1  | 1.288e+1/1.288e+1/1.7e-5 | 55    | 7.4   | 1.288e+1/1.288e+1/3.9e-6 | 177   | 25.4  | 1.288e+1/1.288e+1/4.9e-6 |
| 200  | 0.50     | 79    | 10.5  | 1.755e+1/1.755e+1/1.7e-6 | 20    | 2.8   | 1.755e+1/1.755e+1/4.2e-9 | 93    | 12.9  | 1.755e+1/1.755e+1/2.7e-6 |
| 300  | 0.05     | 45    | 7.8   | 1.259e+1/1.259e+1/1.4e-5 | 38    | 7.2   | 1.259e+1/1.259e+1/6.4e-6 | 224   | 41.0  | 1.259e+1/1.259e+1/1.9e-5 |
| 300  | 0.10     | 227   | 35.6  | 1.768e+1/1.768e+1/1.9e-6 | 43    | 7.5   | 1.768e+1/1.768e+1/1.3e-9 | 95    | 16.3  | 1.768e+1/1.768e+1/1.2e-5 |
| 300  | 0.50     | 76    | 12.2  | 2.253e+1/2.253e+1/1.2e-6 | 20    | 3.5   | 2.253e+1/2.253e+1/2.3e-9 | 111   | 19.1  | 2.253e+1/2.253e+1/2.0e-6 |

5.2 Fused lasso logistic regression problem

As discussed in [21, Section 5], the fused lasso logistic regression problem can be presented as follows:

\[
\min_{z \in \mathbb{R}^n} p_\text{lasso}(z) := \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T \tilde{z} + z_n))) + \lambda_1 \sum_{i=1}^{n-1} |z_i| + \lambda_2 \sum_{i=1}^{n-2} |z_{i+1} - z_i|,
\]

where $a_i \in \mathbb{R}^m$ are samples, $b_i \in \{-1, 1\}$, $i = 1, ..., m$, $m < n$, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are regularization parameters, $z \in \mathbb{R}^n$, $\tilde{z} \in \mathbb{R}^{n-1}$ is the vector that contains the first $n - 1$ entries of $z$ and $z_i$ denotes the $i$th entry of $z$. It is easy to see that (50) is in the form of (1) with

\[
h(z) = \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T \tilde{z} + z_n))),
\]

\[
\mathcal{P}(u) = \lambda_1 \sum_{i=1}^{n-1} |u_i| + \lambda_2 \sum_{i=n}^{2n-3} |u_i|, \quad \mathcal{M} = M := \begin{pmatrix} I_{n-1} \ 0 \\ E_{n-2} \ 0 \end{pmatrix}, \quad b = 0,
\]

where $M$ is a $(2n - 3) \times n$ matrix, $I_{n-1}$ is the identity matrix of dimension $n - 1$ and $E_{n-2}$ is the $(n - 2) \times (n - 1)$ matrix whose diagonal entries are 1, upper diagonal entries are $-1$, and is zero otherwise. To further simplify notations, we define a matrix $A$ whose $ith$ row is given by $(-b_i a_i^T - b_i)$ and a function $l(v) := \sum_{i=1}^m \log(1 + \exp(v_{i})).$ Then $h(z) = l(Az)$. Moreover, it is routine to show that the conjugate function of $l$ is given by

\[
l^*(u) = \sum_{i=1}^m (u_i \log(u_i) + (1 - u_i) \log(1 - u_i)),
\]
with domain $0 \leq u_i \leq 1$ for all $i$. Hence, the dual problem \( (50) \) of \( (51) \) is given by
\[
\min_{\nu, y} d_{\text{lasso}}(\nu) := \sum_{i=1}^{m} (\nu_i \log(\nu_i) + (1 - \nu_i) \log(1 - \nu_i))
\]
\[
s.t. \quad A^T \nu + M^T y = 0, \quad \max_{1 \leq i \leq n-1} |y_i| \leq \lambda_1, \quad \max_{n \leq i \leq 2n-3} |y_i| \leq \lambda_2,
\]
where $A^T$ and $M^T$ are the transpose of the matrices $A$ and $M$, respectively, and $\nu$ is related to the $x$ in \( (9) \) via $x = A^T \nu$. Since $m < n$, we have $\nu = (A^T)^\dagger x$, where $\dagger$ denotes the pseudoinverse. First, as mentioned in the introduction, the proximal mapping of $\nabla h$ is Lipschitz continuous with modulus bounded by 0.25. It then follows immediately from $h(z) = l(Az)$ that $\nabla h$ is Lipschitz continuous with modulus bounded by $0.25 \lambda_{\text{max}}(A^T A)$, the maximum eigenvalue of $A^T A$. Next, notice that
\[
M^T M = \begin{pmatrix}
2 & -1 & 0 & \cdots & \cdots & 0 & 0 \\
-1 & 3 & -1 & \cdots & \cdots & 0 & 0 \\
0 & -1 & 3 & \cdots & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & \cdots & -1 & 3 & -1 & 0 \\
0 & \cdots & \cdots & 0 & -1 & 2 & 0 \\
0 & \cdots & \cdots & 0 & 0 & 0 & 0
\end{pmatrix} \leq \begin{pmatrix}
3 & -1 & 0 & \cdots & \cdots & 0 & 0 \\
-1 & 3 & -1 & \cdots & \cdots & 0 & 0 \\
0 & -1 & 3 & \cdots & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & \cdots & -1 & 3 & -1 & 0 \\
0 & \cdots & \cdots & 0 & -1 & 3 & 0 \\
0 & \cdots & \cdots & 0 & 0 & 0 & 0
\end{pmatrix} =: M_2,
\]
while the maximum eigenvalue of $M_2$ is bounded above by 5; see, for example, [15, Page 25]. Hence, we have $\|M^* M\| \leq 5$. On the other hand, to apply the MFBS method, we need to bound the Lipschitz continuity modulus of $G(z, y)$, where $G(z, y) := \|A^T y + \nabla h(z)\|$. Using [23, Lemma 6.2] and the bound $\|M^* M\| \leq 5$, we see that $L_M$ can be chosen to be $\frac{1}{2} (0.25 \lambda_{\text{max}}(A^T A) + \sqrt{(0.25 \lambda_{\text{max}}(A^T A))^2 + 20})$.

For simplicity, we initialize both algorithms at the origin, i.e., $(y^0, z^0) = (0, 0)$. We terminate the algorithms when
\[
\max \left\{ \frac{\min_{x \in \Gamma} p_{\text{lasso}}(z^*) + d_{\text{lasso}}(\tilde{v}^t)}{\max\{\min_{x \in \Gamma} p_{\text{lasso}}(z^*), 1\}}, \frac{5\|A^T \tilde{v}^t + M^T y^t\|}{\max\{\|A^T \tilde{v}^t\|, \|M^T y^t\|, 1\}} \right\} < tol
\]
for some $tol > 0$ \footnote{For the MFBS method, we used $(u^t, v^t)$ in place of $(z^t, y^t)$ in \( (52) \), and set $x^t$ as the derivative of $h$ at the corresponding point, for all $t$.} where
\[
\tilde{v}^t = \begin{cases} -(A^T)^\dagger M^T y^t & \text{if } 0 \leq -(A^T)^\dagger M^T y^t \leq 1, \\
(A^T)^\dagger x^t & \text{otherwise,}
\end{cases}
\]
the set $\Gamma := \{ s : 1 \leq s \leq t, \ s \text{ is a multiple of } 500 \}$, and the above criterion \( (52) \) is checked every 500 iterations. In our test below, we take $tol = 1e-4$ for both algorithms. Moreover, for our PPG
algorithm, we take $\beta = \frac{1.95}{L} = \frac{7.8}{\lambda_{\text{max}}(A^TA)}$ and $\tau = \frac{39}{\lambda_{\text{max}}(A^TA)}$, while for the MFBS method (6), we take $L_M = \frac{1}{2}(0.25\lambda_{\text{max}}(A^TA)) + \sqrt{(0.25\lambda_{\text{max}}(A^TA))^2 + 20}$ as discussed above, and $\sigma = 0.95$.

We generate random instances for our test. We start by generating an $m \times (n-1)$ matrix $C$ with i.i.d. standard Gaussian entries. We then normalize $C$ so that each column has norm 1 as in [4, Section 4.1]. Moreover, mimicking [21, Section 6.2] and [40, Section 3.1], we take

$$h_j := \begin{cases} 
20\xi_1 & \text{if } j = 1, 2, \ldots, 20, \\
30\xi_2 & \text{if } j = 41, \\
10\xi_3 & \text{if } j = 71, \ldots, 85, \\
20\xi_4 & \text{if } j = 121, \ldots, 125, \\
0 & \text{otherwise,}
\end{cases}$$

where $\xi_i$, $i = 1, \ldots, 4$, are random numbers following standard Gaussian distribution. We then set $b$ to be the sign vector of $C\hat{x} + \xi_5e$, where $\xi_5$ is a random number in $[0, 1]$ and $e$ is the vector of all ones as in [21, Section 6.2]. We next form a matrix $C'$ by multiplying each column of $C$ entrywise with $-b$. The $m \times n$ input matrix $A$ for (50) is then formed as $(C' - b)$. Furthermore, we set $\lambda_1 = \alpha m$ and $\lambda_2 = 100\lambda_1$ for some $\alpha > 0$, mimicking the choice in [21, Section 6.2].

We fix $m = 250$. For each $n = 10000, 20000, 30000$ and $\alpha = 1e-4, 3e-4$ and $5e-4$, we generate 10 instances as described above. The computational results, averaged over the 10 instances, are reported in Table 2 where we report the number of iterations ($\text{iter}$), CPU time in seconds ($\text{cpu}$), primal objective value (50) at termination ($\text{pobj}$), dual objective value (51) at termination ($\text{dobj}$) and the relative dual infeasibility

$$\frac{\|A^T\hat{p}^t + M^T y^t\|}{\max(\|A^T\hat{p}^t\|, \|M^T y^t\|, 1)}$$

at termination ($\text{dfeas}$). We only report the results for our algorithm since the MFBS method (6) never terminates within 20000 iterations. From Table 2, we observe that our algorithm performs reasonably well, and tends to be slower when $\alpha$ is smaller.

Table 2: Results for PPG algorithm on solving (50)

| $n$   | $\alpha$ | $\text{iter}$ | $\text{cpu}$ | $\text{pobj/dobj/dfeas}$ |
|-------|----------|---------------|--------------|--------------------------|
| 10000 | 1e-4     | 6500          | 18.6         | 1.167e+2/1.167e+2/1.1e-5 |
| 10000 | 3e-4     | 2400          | 6.8          | 1.56e+2/1.56e+2/8.5e-6   |
| 10000 | 5e-4     | 1500          | 4.2          | 1.67e+2/1.67e+2/7.2e-6   |
| 20000 | 1e-4     | 5750          | 44.8         | 1.262e+2/1.262e+2/1.3e-5 |
| 20000 | 3e-4     | 3050          | 23.7         | 1.491e+2/1.491e+2/7.8e-6 |
| 20000 | 5e-4     | 1600          | 12.1         | 1.631e+2/1.631e+2/4.4e-6 |
| 30000 | 1e-4     | 8200          | 91.8         | 1.14e+2/1.14e+2/1.1e-5   |
| 30000 | 3e-4     | 2950          | 33.3         | 1.504e+2/1.504e+2/6.6e-6 |
| 30000 | 5e-4     | 1800          | 21.4         | 1.652e+2/1.652e+2/9.7e-6 |

6 Concluding remarks

We have proposed a new algorithm for solving (1) which admits easy subproblems assuming the proximal mappings of $\tau P$, $\tau > 0$, are easy to compute. Our algorithm reduces to the usual proximal gradient algorithm when the affine map is just the identity map, and is equivalent to applying a proximal AMA to the dual of (1). We established global convergence and discussed iteration

\footnote{We experimented with $\beta = \frac{4.05}{L}$, $\frac{1}{L}$ and $\frac{1.05}{L}$. Setting $\beta = \frac{4.05}{L}$ seems to work best for all values of $\alpha$.}
complexity. Our computational results on solving nuclear norm regularized system realization problem and the fused lasso logistic regression problem show that our algorithm works reasonably well on large-scale instances.

Acknowledgements. The author would like to thank Christopher Jordan-Squire for carefully proofreading an early version of this manuscript.

References

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

[2] S. Becker, E. J. Candès and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Comput.* 3, pp. 165–218 (2011).

[3] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, 2nd edition (2006).

[4] E. J. Candès and T. Tao. The Dantzig selector: statistical estimation when \( p \) is much larger than \( n \). *Ann. Stat.* 35, pp. 2313–2351 (2007).

[5] J. Eckstein. Some saddle-function splitting methods for convex programming. *Optim. Method Softw.* 4, pp. 75–83 (1994).

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

[7] M. Elad, P. Milanfar and R. Rubinstein. Analysis versus synthesis in signal priors *Inverse Probl.* 23, pp. 947–968 (2007).

[8] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, eds., *Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems*. North-Holland, Amsterdam (1983).

[9] M. Fazel, T. K. Pong, D. Sun and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. To appear in *SIAM J. Matrix Anal. A.* (2013).

[10] M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *Int. J. Syst. Sci.* 12, pp. 989–1000 (1981).

[11] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, eds., *Augmented Lagrangion Methods: Applications to the Solution of Boundary Problems*. North-Holland, Amsterdam, 1983.

[12] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Comput. Math. Appl.* 2, pp. 17–40 (1976).

[13] R. Glowinski and A. Marroco. Sur l’approximation, par elements finis d’ordre un, et la resolution, par penalisation-dualit’e, d’une classe de problemes de Dirichlet non lineares. *Revue Francaise d’Automatique, Informatique et Recherche Opérationnelle.* 9 (R-2), pp. 41–76 (1975).

[14] B. He, L. Liao, D. Han and H. Yang. A new inexact alternating directions method for monotone variational inequalities. *Math. Program.* 92, pp. 103–118 (2002).
[15] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 2nd edition (2002).

[16] G. Lan, Z. Lu and R. D. C. Monteiro. Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming. *Math. Program.* 126, pp. 1–29 (2011).

[17] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM. J. Matrix Anal. A.* 31, pp. 1235–1256 (2009).

[18] Z. Liu and L. Vandenberghe. Semidefinite programming methods for system realization and identification. *Proc. 48th IEEE Conference on Decision and Control*, pp. 4676–4681 (2009).

[19] Z. Liu, A. Hansson and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. To appear in *Syst. Control Lett.* (2013).

[20] S. Ma, D. Goldfarb and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* 128, pp. 321–353 (2011).

[21] S. Ma and S. Zhang. An extragradient-based alternating direction method for convex minimization. Preprint, Jan 2013. Available at [http://arxiv.org/abs/1301.6308](http://arxiv.org/abs/1301.6308).

[22] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. *Proc. American Control Conference* (2010).

[23] R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of Tseng’s modified F-B splitting and Korpelevich’s methods for hemi-variational inequalities with applications to saddle point and convex optimization problems. *SIAM J. Optim.* 21, pp. 1688–1720 (2011).

[24] Y. Nesterov. A method for solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.* 27(2), pp. 372–376 (1983).

[25] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers (2003).

[26] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.* 16, pp. 235–249 (2005).

[27] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.* 103, pp. 127–152 (2005).

[28] M. Patriksson. Cost approximation: a unified framework of descent algorithms for nonlinear programs. *SIAM J. Optim.* 8, pp. 561–582 (1998).

[29] T. K. Pong, P. Tseng, S. Ji and J. Ye. Trace norm regularization: reformulations, algorithms, and multi-task learning. *SIAM J. Optim.* 20, pp. 3465–3489 (2010).

[30] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton (1970).

[31] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pac. J. Optim.* 6, pp. 615–640 (2010).

[32] M. W. Schmidt, N. Le Roux and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS*, pp. 1458–1466 (2011).

[33] R. Tomioka, T. Suzuki and M. Sugiyama. Augmented Lagrangian methods for learning, selecting, and combining features. In S. Sra, S. Nowozin and S. J. Wright (Eds.), *Optimization for Machine Learning*. MIT Press (2012).
[34] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.* 29, pp. 119–138 (1991).

[35] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report (2008).

[36] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Program.* 125, pp. 263–295 (2010).

[37] L. Vandenberghe. Convex optimization techniques in system identification. *Proc. IFAC Symposium on System Identification*, pp. 71–76 (2012).

[38] M. H. Xu and T. Wu. A class of linearized proximal alternating direction methods. *J. Optim. Theory Appl.* 151, pp. 321–337 (2011).

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

[40] G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Comput. Stat. Data An.* 55, pp. 1552–1569 (2011).

[41] X. Zhang, M. Burger and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.* 46, pp. 20–46 (2011).