

The proximal-proximal gradient algorithm

Ting Kei Pong *

August 23, 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 τP , $\tau > 0$, 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 the stepsizes are suitably chosen, and it is equivalent to applying a variant of the alternating minimization algorithm [35] to the dual problem. Moreover, it is closely related to inexact proximal gradient algorithms [29, 33]. 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

$$v_{\text{opt}} := \min_z F(z) := h(z) + P(\mathcal{M}z - b), \quad (1)$$

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, \mathcal{M} is a nonzero linear map between two finite dimensional Hilbert spaces \mathcal{Z} and \mathcal{Y} , and b is a given element in \mathcal{Y} . We assume throughout that (1) has an optimal solution. We also assume the following constraint qualification to rule out degenerate instances:

$$(\text{Range}(\mathcal{M}) - b) \cap \text{ri}(\text{dom } P) \neq \emptyset; \quad (2)$$

see Section 1.1 for notations and definitions. This condition is trivially satisfied if P is continuous or \mathcal{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 τP for any $\tau > 0$.¹

*Department of Computer Science, University of British Columbia, Vancouver, BC, Canada V6T 1Z4. E-mail: tkpong@cs.ubc.ca. This author is a PIMS postdoctoral fellow.

¹Here and throughout the paper, by “easy to compute”, we mean an easy, and preferably a closed form solution, is available.

Problem (1) arises from various applications, with h usually the loss function, P the regularization function and $\mathcal{M}z - b$ modeling some priors known about the desired solution. One example arises from the maximum a posteriori probability analysis approach in signal processing [8], where $b = 0$, P is the ℓ_1 norm and \mathcal{M} is the so-called analyzing operator, with a higher value of $P(\mathcal{M}z)$ 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 $\mathcal{M}z$ is a block Hankel matrix of suitable size; see, for example, [10, 18, 19, 20, 38]. Finally, we also mention the fused lasso logistic regression problem introduced recently in [22] in which h is the smooth logistic loss function, $b = 0$, P is the ℓ_1 norm and \mathcal{M} is a suitably defined linear map. We refer the readers to [22, Section 5] for more details about this particular problem. Notice that in all the above applications, the proximal mapping of P and those of τP , $\tau > 0$ are computable via a closed form formula; see, for example, [34, Section 9.2].

Since h is differentiable with Lipschitz continuous gradient and P is nonsmooth, it is natural to consider a proximal gradient algorithm [11] for solving (1), where one updates

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

For such an algorithm, it is well-known that $F(z^t) - F(z^*) = O(\frac{1}{t})$; see, for example, [26, Theorem 2.1.14]. Furthermore, the whole sequence generated is convergent to an optimal solution of (1); see, for example, [35, Section 2]. This simple algorithm, together with its accelerated versions using Nesterov extrapolation techniques [25, 26, 27, 28], has recently been studied extensively in the literature; see, for example, [1, 2, 10, 17, 21, 23, 30, 32, 37]. 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 $\frac{1}{L}P(\mathcal{M} \cdot - b)$. While the proximal mapping of $\frac{1}{L}P$ is easy to compute, the proximal mapping of the composition $\frac{1}{L}P(\mathcal{M} \cdot - b)$, 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

$$\begin{aligned} \min_{z, u} \quad & h(z) + P(u) \\ \text{s.t.} \quad & u = \mathcal{M}z - b, \end{aligned} \quad (4)$$

and apply a suitable algorithm, for example, the alternating direction method of multipliers (ADMM); see, for example, [6, 7, 9, 12, 13, 14]. This algorithm consists of the following updates

$$\begin{cases} z^{t+1} \in \text{Arg} \min_z \left\{ h(z) + \langle y^t, \mathcal{M}z \rangle + \frac{\beta}{2} \|u^t - \mathcal{M}z + b\|^2 \right\}, \\ u^{t+1} = \arg \min_u \left\{ P(u) - \langle y^t, u \rangle + \frac{\beta}{2} \|u - \mathcal{M}z^{t+1} + b\|^2 \right\}, \\ y^{t+1} = y^t - \gamma \beta (u^{t+1} - \mathcal{M}z^{t+1} + b), \end{cases}$$

where $\beta > 0$ and $\gamma \in (0, \frac{\sqrt{5}+1}{2})$, and we use capital Arg min to indicate that the minimizer may not be unique. 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 not easy 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, [42, 45]. However, the proximal mapping of $\frac{1}{\beta}h$ can still be difficult to compute. In the recent work [22], 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 (4).

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:

$$\min_z \max_y \{h(z) + \langle y, \mathcal{M}z - b \rangle - P^*(y)\}, \quad (5)$$

where P^* is the convex conjugate of P . One can then apply a suitable algorithm, for example, Tseng’s modified forward backward splitting (MFBS) method proposed in [36], which consists of the following updates in each iteration (see [36, Example 5])

$$\begin{cases} v^t = \arg \min_y \left\{ P^*(y) + \langle b - \mathcal{M}z^t, y \rangle + \frac{L_{\mathcal{M}}}{2\sigma} \|y - y^t\|^2 \right\}, & u^t = z^t - \frac{\sigma}{L_{\mathcal{M}}} (\nabla h(z^t) + \mathcal{M}^* y^t), \\ y^{t+1} = v^t - \frac{\sigma}{L_{\mathcal{M}}} (\mathcal{M}z^t - \mathcal{M}u^t), & z^{t+1} = u^t - \frac{\sigma}{L_{\mathcal{M}}} (\nabla h(u^t) + \mathcal{M}^* v^t - \nabla h(z^t) - \mathcal{M}^* y^t), \end{cases} \quad (6)$$

where $\sigma \in (0, 1)$ and $L_{\mathcal{M}}$ is an upper bound of the Lipschitz continuity modulus of the function

$$G(z, y) = \begin{pmatrix} \nabla h(z) + \mathcal{M}^* y \\ b - \mathcal{M}z \end{pmatrix},$$

and \mathcal{M}^* is the adjoint linear map of \mathcal{M} . The subproblem in (6) can be shown to be equivalent to a computation of the proximal mapping of $\frac{L_{\mathcal{M}}}{\sigma} P$, and it was further shown in [36, Theorem 3.4(b)] 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 [35] (AMA) to the dual problem of (1). It reduces to the proximal gradient algorithm if \mathcal{M} is the identity map, $b = 0$ and the stepsizes are suitably chosen; and is closely related to inexact proximal gradient algorithms [29, 33]. 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, [19]) against the primal ADMM2 in [10] and the MFBS method in [36], as well as on the fused lasso logistic regression problem (see [22]) against the MFBS method in [36]. 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 \mathcal{X} , \mathcal{Y} and \mathcal{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 \mathcal{X} , \mathcal{Y} and \mathcal{Z}). For a linear map \mathcal{A} , \mathcal{A}^* denotes the adjoint linear map with respect to the inner product, $\text{Range}(\mathcal{A})$ denotes the image of \mathcal{A} and $\|\mathcal{A}\|$ is the induced operator norm of \mathcal{A} . A linear self-map $\mathcal{T} : \mathcal{Z} \rightarrow \mathcal{Z}$ 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 $z \in \mathcal{Z}$.

We let $\mathcal{T} \succeq 0$ (resp., $\mathcal{T} \succ 0$) 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 $z \in \mathcal{Z}$. We denote the identity map by \mathcal{I} .

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

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

For a proper closed convex function $f : \mathcal{Z} \rightarrow (-\infty, \infty]$, 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 [31, Theorem 12.2] that

$$f^{**} = f. \tag{7}$$

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 (unique) minimizer. Recall from [31, 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:

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

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 $\tau P(\mathcal{M} \cdot -b)$, $\tau > 0$, 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 [35, Section 2].

To describe the equivalence, we note from (2) and [31, Theorem 31.2] that

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

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

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

Since ∇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 [35, Section 4] to

solve the dual problem (9):

$$\begin{cases} x^{t+1} = \arg \min_x \{h^*(x) - \langle z^t, x \rangle\}, \\ y^{t+1} \in \text{Arg} \min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \|x^{t+1} + \mathcal{M}^* y\|^2 \right\}, \\ z^{t+1} = z^t - \beta(x^{t+1} + \mathcal{M}^* y^{t+1}). \end{cases} \quad (10)$$

This algorithm is known to converge if $\beta \in (0, \frac{2}{L})$, assuming that the iterates are well-defined; see [35, Proposition 3] for the precise assumptions. Moreover, with $\beta = \frac{1}{L}$, it can be shown (see [35, Section 2]) that this algorithm is equivalent to the proximal gradient algorithm (3), 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

$$x^{t+1} = \arg \min_x \{h^*(x) - \langle z^t, x \rangle\} \Leftrightarrow z^t \in \partial h^*(x^{t+1}) \Leftrightarrow x^{t+1} = \nabla h(z^t), \quad (11)$$

where the last equivalence follows from [31, Theorem 23.5]. On the other hand, from the second and third subproblem of (10), we see that

$$\begin{aligned} y^{t+1} &\in \text{Arg} \min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^* y \rangle + \frac{\beta}{2} \|x^{t+1} + \mathcal{M}^* y\|^2 \right\} \\ &\Leftrightarrow \mathcal{M}z^{t+1} - b \in \partial P^*(y^{t+1}) \Leftrightarrow y^{t+1} \in \partial P(\mathcal{M}z^{t+1} - b) \\ &\Rightarrow \mathcal{M}^* y^{t+1} \in \mathcal{M}^* \partial P(\mathcal{M}z^{t+1} - b) \Rightarrow \mathcal{M}^* y^{t+1} \in \partial P(\mathcal{M} \cdot -b)(z^{t+1}), \end{aligned} \quad (12)$$

where the second equivalence follows from [31, Theorem 23.5] and the last implication follows from [31, Theorem 23.9]. From these it is easy to verify that $\{z^t\}$ generated from the AMA satisfies (3). Conversely, if $\{z^t\}$ is generated according to (3), then from the first-order optimality condition, (2) and the subdifferential calculus rules [31, Theorem 23.8], [31, 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 (3). 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 [45] 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 (1) 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 [35, Section 4]. This method aims at solving optimization problems in the following form:

$$\begin{aligned} \min_{x,y} \quad & f(x) + g(y) \\ \text{s.t.} \quad & \mathcal{A}x + \mathcal{B}y = c, \end{aligned} \quad (13)$$

where $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Z}$ and $\mathcal{B} : \mathcal{Y} \rightarrow \mathcal{Z}$ are linear maps on the finite dimensional Hilbert spaces \mathcal{X} , \mathcal{Y} and \mathcal{Z} , $c \in \mathcal{Z}$, $f : \mathcal{X} \rightarrow (-\infty, +\infty]$ and $g : \mathcal{Y} \rightarrow (-\infty, +\infty]$ 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

$$\langle u_1 - u_2, x_1 - x_2 \rangle \geq \|x_1 - x_2\|_{\Sigma_f}^2; \quad (14)$$

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

$$\langle v_1 - v_2, y_1 - y_2 \rangle \geq \|y_1 - y_2\|_{\Sigma_g}^2, \quad (15)$$

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 $\mathcal{T} = 0$ and $\gamma = 1$.

Proximal AMA for (13)

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

Step 1. Set

$$\begin{cases} x^{t+1} = \arg \min_x \{f(x) - \langle z^t, \mathcal{A}x \rangle\}, \\ y^{t+1} = \arg \min_y \left\{ g(y) - \langle z^t, \mathcal{B}y \rangle + \frac{\beta}{2} \|\mathcal{A}x^{t+1} + \mathcal{B}y - c\|^2 + \frac{1}{2} \|y - y^t\|_{\mathcal{T}}^2 \right\}, \\ z^{t+1} = z^t - \gamma\beta(\mathcal{A}x^{t+1} + \mathcal{B}y^{t+1} - c), \end{cases} \quad (16)$$

where $\beta > 0$, $\gamma > 0$ 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 $\frac{1}{\beta}g$, as discussed recently in [45] in the settings of ADMM. This would be particularly useful when the proximal mapping of $\frac{1}{\beta}g$ is easy to compute. Moreover, as discussed in [10, 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 [9] for the convergence of the alternating direction method of multipliers (ADMM), a closely related algorithm. See also [12, Section 6], [35, Appendix A], [41] and [10, Theorem B.1]. We also note that the convergence of AMA without the proximal term (i.e., $\mathcal{T} = 0$), $\gamma = 1$ and with a varying step size β was established in [35, Proposition 3], under slightly different assumptions from below.

Before stating the theorem, we make the following assumption:

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

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

Note that for any $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$ and $\bar{z} \in \mathcal{Z}$ 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 [31, Corollary 28.2.2]), with (17) being satisfied by

any optimal solution (\bar{x}, \bar{y}) to problem (13) (see [31, Corollary 28.3.1]) and any optimal solution \bar{z} to the dual problem of (13) (see [31, Corollary 28.4.1]).²

Theorem 1. *Suppose that **A1** holds and let $\{(x^t, y^t, z^t)\}$ be generated from the proximal AMA. Suppose further that $\beta > 0$, $\mathcal{T} \succeq 0$ and $\gamma > 0$ are chosen so that*

- (i) $\Sigma_g + \mathcal{T} + \beta\mathcal{B}^*\mathcal{B} \succ 0$;
- (ii) for some $\mu > 0$, $2\Sigma_f - (\beta + \mu)\mathcal{A}^*\mathcal{A} \succ 0$ and $\gamma < 1 + \frac{\min\{\beta, \mu\}}{2\beta}$.

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. We first observe from $\Sigma_f \succ 0$ and assumption (i) of the theorem that the iterates are all well-defined. Furthermore, the iterates of the proximal AMA can be equivalently rewritten as follows:

$$\begin{cases} 0 \in \partial f(x^{t+1}) - \mathcal{A}^*z^t, \\ 0 \in \partial g(y^{t+1}) - \mathcal{B}^*(z^t - \beta(\mathcal{A}x^{t+1} + \mathcal{B}y^{t+1} - c)) + \mathcal{T}(y^{t+1} - y^t), \\ z^{t+1} = z^t - \gamma\beta(\mathcal{A}x^{t+1} + \mathcal{B}y^{t+1} - c). \end{cases} \quad (18)$$

For notational simplicity, we fix any $(\bar{x}, \bar{y}, \bar{z})$ satisfying (17) and write $w_e^t := w^t - \bar{w}$, where w represents x , y and z respectively. From (18), (17), (14) and (15), we have

$$\langle \mathcal{A}^*z_e^t, x_e^{t+1} \rangle \geq \|x_e^{t+1}\|_{\Sigma_f}^2, \quad \langle \mathcal{B}^*(z_e^t - \beta(\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1})) - \mathcal{T}(y^{t+1} - y^t), y_e^{t+1} \rangle \geq \|y_e^{t+1}\|_{\Sigma_g}^2. \quad (19)$$

Summing the above two inequalities and rearranging terms, we obtain

$$\langle z_e^t, \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1} \rangle - \beta \langle \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}, \mathcal{B}y_e^{t+1} \rangle - \langle \mathcal{T}(y^{t+1} - y^t), y_e^{t+1} \rangle \geq \|x_e^{t+1}\|_{\Sigma_f}^2 + \|y_e^{t+1}\|_{\Sigma_g}^2.$$

Using the relation $\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1} = (\gamma\beta)^{-1}(z^t - z^{t+1})$ to the first term on the left, we see that

$$\frac{1}{\gamma\beta} \langle z_e^t, z^t - z^{t+1} \rangle - \beta \langle \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}, \mathcal{B}y_e^{t+1} \rangle - \langle y^{t+1} - y^t, y_e^{t+1} \rangle_{\mathcal{T}} \geq \|x_e^{t+1}\|_{\Sigma_f}^2 + \|y_e^{t+1}\|_{\Sigma_g}^2.$$

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 $\langle u, v \rangle_{\mathcal{T}} = \frac{1}{2}(\|u\|_{\mathcal{T}}^2 + \|v\|_{\mathcal{T}}^2 - \|u - v\|_{\mathcal{T}}^2)$ to the third term on the left, multiplying both sides by a factor of 2 and rearranging terms, we arrive at

$$\begin{aligned} & \left(\frac{1}{\gamma\beta} \|z_e^t\|^2 + \|y_e^t\|_{\mathcal{T}}^2 \right) - \left(\frac{1}{\gamma\beta} \|z_e^{t+1}\|^2 + \|y_e^{t+1}\|_{\mathcal{T}}^2 \right) \\ & \geq 2\|x_e^{t+1}\|_{\Sigma_f}^2 + 2\|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \beta\|\mathcal{B}y_e^{t+1}\|^2 - \beta\|\mathcal{A}x_e^{t+1}\|^2 + \left(\frac{1}{\gamma^2\beta} - \frac{1}{\gamma\beta} \right) \|z^{t+1} - z^t\|^2 \\ & = \|x_e^{t+1}\|_{2\Sigma_f - \beta\mathcal{A}^*\mathcal{A}}^2 + 2\|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \beta\|\mathcal{B}y_e^{t+1}\|^2 + \left(\frac{1}{\gamma^2\beta} - \frac{1}{\gamma\beta} \right) \|z^{t+1} - z^t\|^2 \\ & = \|x_e^{t+1}\|_{\mathcal{S}}^2 + 2\|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \mu\|\mathcal{A}x_e^{t+1}\|^2 + \beta\|\mathcal{B}y_e^{t+1}\|^2 + \left(\frac{1}{\gamma^2\beta} - \frac{1}{\gamma\beta} \right) \|z^{t+1} - z^t\|^2 \\ & \geq \|x_e^{t+1}\|_{\mathcal{S}}^2 + 2\|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + (\beta - \alpha)\|\mathcal{B}y_e^{t+1}\|^2 + \left(\frac{1}{\gamma^2\beta} - \frac{1}{\gamma\beta} \right) \|z^{t+1} - z^t\|^2 \\ & \quad + \frac{\min\{\alpha, \mu\}}{2} \|\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}\|^2 \\ & = \|x_e^{t+1}\|_{\mathcal{S}}^2 + 2\|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + (\beta - \alpha)\|\mathcal{B}y_e^{t+1}\|^2 + \frac{1}{\gamma^2\beta} \left(\frac{\min\{\alpha, \mu\}}{2\beta} + 1 - \gamma \right) \|z^{t+1} - z^t\|^2, \end{aligned} \quad (20)$$

²We note here that it is very common in the literature to use a constraint qualification in place of **A1** as the standing assumption for a convergence result. However, since we will soon apply this proximal AMA to the dual problem (9) of (1), we chose to use **A1**, which is a condition readily shown to hold for (9) under our assumptions on (1); see Theorem 3.

where $\mathcal{S} = 2\Sigma_f - (\beta + \mu)\mathcal{A}^*\mathcal{A}$, $\alpha > 0$ and we used the relation $\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1} = (\gamma\beta)^{-1}(z^t - z^{t+1})$ in the first inequality and the last equality. We further choose $\alpha > 0$ in (20) so that $\alpha < \beta$ and

$$\gamma < \frac{\min\{\alpha, \mu\}}{2\beta} + 1;$$

this is possible due to assumption (ii) of the theorem. From this choice and (20), it follows that the sequence $\{\frac{1}{\gamma\beta}\|z_e^t\|^2 + \|y_e^t\|_{\mathcal{T}}^2\}$ is monotonically nonincreasing (and so, bounded), and

$$\lim_{t \rightarrow \infty} \|x_e^{t+1}\|_{\mathcal{S}}^2 + \|y_e^{t+1}\|_{\Sigma_g}^2 + \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \|\mathcal{B}y_e^{t+1}\|^2 + \|z^{t+1} - z^t\|^2 = 0. \quad (21)$$

These together with the positive definiteness of $\Sigma_g + \mathcal{T} + \beta\mathcal{B}^*\mathcal{B}$ and \mathcal{S} establish the boundedness of $\{(x^t, y^t, z^t)\}$. It then follows immediately that there exists a subsequence $\{(x^{t_i}, y^{t_i}, z^{t_i})\}$ that converges to a limit point $(x^\diamond, y^\diamond, z^\diamond)$. We next show that $(x^\diamond, y^\diamond, z^\diamond)$ satisfies (17).

To see this, we first observe from (21) and the positive definiteness of \mathcal{S} that

$$\lim_{t \rightarrow \infty} \|x_e^{t+1}\| = 0, \quad \lim_{t \rightarrow \infty} \|y_e^t\|_{\Sigma_g} + \|\mathcal{B}y_e^t\| = 0, \quad \lim_{t \rightarrow \infty} \|y^{t+1} - y^t\|_{\mathcal{T}} = \lim_{t \rightarrow \infty} \|z^{t+1} - z^t\| = 0. \quad (22)$$

Taking limits on both sides of (18) along the subsequence $\{(x^{t_i-1}, y^{t_i-1}, z^{t_i-1})\}$, using the third relation in (22) and the closedness of the graphs of ∂f and ∂g [3, Page 80], we conclude that

$$\mathcal{A}^*z^\diamond \in \partial f(x^\diamond), \quad \mathcal{B}^*z^\diamond \in \partial g(y^\diamond), \quad \mathcal{A}x^\diamond + \mathcal{B}y^\diamond - c = 0,$$

showing that $(x^\diamond, y^\diamond, z^\diamond)$ satisfies (17).

To complete the proof, now it remains to show that $(x^\diamond, y^\diamond, z^\diamond)$ is the unique limit of $\{(x^t, y^t, z^t)\}$. Since $(x^\diamond, y^\diamond, z^\diamond)$ satisfies (17), we could replace $(\bar{x}, \bar{y}, \bar{z})$ with $(x^\diamond, y^\diamond, z^\diamond)$ in the above arguments, starting from (19). First, the convergence of $\{x^t\}$ follows immediately from (22). Next, notice that the subsequence $\{\frac{1}{\gamma\beta}\|z_e^{t_i}\|^2 + \|y_e^{t_i}\|_{\mathcal{T}}^2\}$ converges to 0 as $i \rightarrow \infty$. Since this sequence is also non-increasing, we must have

$$\lim_{t \rightarrow \infty} \frac{1}{\gamma\beta}\|z_e^t\|^2 + \|y_e^t\|_{\mathcal{T}}^2 = 0. \quad (23)$$

From this, we see immediately that $\lim_{t \rightarrow \infty} z^t = z^\diamond$. Finally, using (23), (22) and the assumption that $\Sigma_g + \mathcal{T} + \beta\mathcal{B}^*\mathcal{B} \succ 0$, we obtain further that $\lim_{t \rightarrow \infty} y^t = y^\diamond$. This completes the proof. \square

Next, we explore the iteration complexity of the above algorithm. Obviously, one can characterize optimality of a point (\hat{x}, \hat{y}) by

$$f(\hat{x}) + g(\hat{y}) = f(\bar{x}) + g(\bar{y}) \quad \text{and} \quad \|\mathcal{A}\hat{x} + \mathcal{B}\hat{y} - c\| = 0,$$

where (\bar{x}, \bar{y}) is a solution to (13). In the next theorem, we study the iteration complexity of the proximal AMA by establishing upper bound on $\|\mathcal{A}x + \mathcal{B}y - c\|$ along a suitable sequence, and upper and lower bounds on the difference between the function value and the optimal value along the same sequence. Our proof technique is similar to [40, Lemma 2] which established bounds similar to (24) and (25) 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 $\beta > 0$, $\mathcal{T} \succeq 0$ and $\gamma > 0$ are chosen so that*

- (i) $\Sigma_g + \mathcal{T} + \beta\mathcal{B}^*\mathcal{B} \succ 0$;
- (ii) *for some positive μ , δ and σ , $2\Sigma_f - (\beta + \mu)\mathcal{A}^*\mathcal{A} \succeq \delta I$ and $\gamma + \sigma \leq 1 + \frac{\min\{\beta, \mu\}}{2\beta}$.*

Define

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

Then it holds that

$$\begin{aligned} & -\frac{\|\bar{z}\|}{\sqrt{N\sigma\beta}} \sqrt{\frac{1}{\gamma\beta} \|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\mathcal{T}}^2} \leq f(\bar{x}^N) + g(\bar{y}^N) - f(\bar{x}) - g(\bar{y}) \\ & \leq \frac{1}{2N} \left(\frac{1}{\gamma\beta} \|z^0\|^2 + \|y^0 - \bar{y}\|_{\mathcal{T}}^2 \right) + \frac{1}{N} \left(\frac{\beta\|\mathcal{A}\|^2}{2\delta} + \frac{\max\{\gamma - 1, 0\}}{2\sigma} \right) \left(\frac{1}{\gamma\beta} \|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\mathcal{T}}^2 \right). \end{aligned} \quad (24)$$

and

$$\|\mathcal{A}\bar{x}^N + \mathcal{B}\bar{y}^N - c\| \leq \frac{1}{\sqrt{N\sigma\beta}} \sqrt{\frac{1}{\gamma\beta} \|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\mathcal{T}}^2}, \quad (25)$$

where $(\bar{x}, \bar{y}, \bar{z})$ is the limit of $\{(x^t, y^t, z^t)\}$ guaranteed by Theorem 1.

Proof. As before, let $(\bar{x}, \bar{y}, \bar{z})$ be the limit of $\{(x^t, y^t, z^t)\}$, and write $w_e^t := w^t - \bar{w}$, where w represents x, y and z respectively. Then it holds that

$$\begin{aligned} & \langle \mathcal{A}^*(z^{t+1} - z^t), x_e^{t+1} \rangle + \langle \mathcal{T}(y^{t+1} - y^t) + (1 - \gamma)\beta\mathcal{B}^*(\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}), y_e^{t+1} \rangle \\ & = \frac{1}{\gamma\beta} \langle z^{t+1} - z^t, \gamma\beta\mathcal{A}x_e^{t+1} \rangle + \langle y^{t+1} - y^t, y_e^{t+1} \rangle_{\mathcal{T}} + (1 - \gamma)\beta \langle \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}, \mathcal{B}y_e^{t+1} \rangle \\ & = \frac{1}{2\gamma\beta} \|z^{t+1} - z^t + \gamma\beta\mathcal{A}x_e^{t+1}\|^2 - \frac{1}{2\gamma\beta} \|z^{t+1} - z^t\|^2 - \frac{\gamma\beta}{2} \|\mathcal{A}x_e^{t+1}\|^2 \\ & \quad + \frac{1}{2} \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \frac{1}{2} \|y_e^{t+1}\|_{\mathcal{T}}^2 - \frac{1}{2} \|y_e^t\|_{\mathcal{T}}^2 + \frac{(1 - \gamma)\beta}{2} (\|\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}\|^2 + \|\mathcal{B}y_e^{t+1}\|^2 - \|\mathcal{A}x_e^{t+1}\|^2) \\ & = \frac{\beta}{2} \|\mathcal{B}y_e^{t+1}\|^2 - \frac{1}{2\gamma\beta} \|z^{t+1} - z^t\|^2 - \frac{\beta}{2} \|\mathcal{A}x_e^{t+1}\|^2 \\ & \quad + \frac{1}{2} \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \frac{1}{2} \|y_e^{t+1}\|_{\mathcal{T}}^2 - \frac{1}{2} \|y_e^t\|_{\mathcal{T}}^2 + \frac{(1 - \gamma)\beta}{2} \|\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}\|^2 \\ & = \frac{\beta}{2} \|\mathcal{B}y_e^{t+1}\|^2 + \frac{1}{2\gamma\beta} \left(\frac{1 - \gamma}{\gamma} - 1 \right) \|z^{t+1} - z^t\|^2 - \frac{\beta}{2} \|\mathcal{A}x_e^{t+1}\|^2 + \frac{1}{2} \|y^{t+1} - y^t\|_{\mathcal{T}}^2 + \frac{1}{2} \|y_e^{t+1}\|_{\mathcal{T}}^2 - \frac{1}{2} \|y_e^t\|_{\mathcal{T}}^2 \end{aligned} \quad (26)$$

where we made use of the elementary relations $\langle u, v \rangle = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|u - v\|^2) = \frac{1}{2}(\|u + v\|^2 - \|u\|^2 - \|v\|^2)$ and $\langle u, v \rangle_{\mathcal{T}} = \frac{1}{2}(\|u\|_{\mathcal{T}}^2 + \|v\|_{\mathcal{T}}^2 - \|u - v\|_{\mathcal{T}}^2)$ in the second equality, the fact that $-\mathcal{A}\bar{x} = \mathcal{B}\bar{y} - c$ in the third equality and the definition of z^{t+1} in the last equality. Moreover, note from (18) that the iterates of the proximal AMA satisfy the following relations:

$$\begin{cases} -\mathcal{A}^*(z^{t+1} - z^t) \in \partial f(x^{t+1}) - \mathcal{A}^*z^{t+1}, \\ -(1 - \gamma)\beta\mathcal{B}^*(\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}) - \mathcal{T}(y^{t+1} - y^t) \in \partial g(y^{t+1}) - \mathcal{B}^*z^{t+1}, \\ -\frac{1}{\gamma\beta}(z^{t+1} - z^t) = \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1} - c. \end{cases} \quad (27)$$

Using this, (26), and the convexity of the Lagrangian function $\ell(x, y, z) := f(x) + g(y) - \langle z, \mathcal{A}x + \mathcal{B}y - c \rangle$ in the variables x and y , we obtain further that

$$\begin{aligned} & \frac{1}{2\gamma\beta} \left(\frac{1 - \gamma}{\gamma} - 1 \right) \|z^{t+1} - z^t\|^2 - \frac{\beta}{2} \|\mathcal{A}x_e^{t+1}\|^2 + \frac{1}{2} \|y_e^{t+1}\|_{\mathcal{T}}^2 - \frac{1}{2} \|y_e^t\|_{\mathcal{T}}^2 \\ & \leq \langle -\mathcal{A}^*(z^{t+1} - z^t), \bar{x} - x^{t+1} \rangle + \langle -(1 - \gamma)\beta\mathcal{B}^*(\mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1}) - \mathcal{T}(y^{t+1} - y^t), \bar{y} - y^{t+1} \rangle \\ & \leq \ell(\bar{x}, \bar{y}, z^{t+1}) - \ell(x^{t+1}, y^{t+1}, z^{t+1}) = f(\bar{x}) + g(\bar{y}) - f(x^{t+1}) - g(y^{t+1}) + \langle z^{t+1}, \mathcal{A}x_e^{t+1} + \mathcal{B}y_e^{t+1} \rangle \\ & = f(\bar{x}) + g(\bar{y}) - f(x^{t+1}) - g(y^{t+1}) + \frac{1}{2\gamma\beta} \|z^t\|^2 - \frac{1}{2\gamma\beta} \|z^{t+1}\|^2 - \frac{1}{2\gamma\beta} \|z^{t+1} - z^t\|^2, \end{aligned}$$

where we made use of the third relation in (27) and the relation $\langle u, v \rangle = \frac{1}{2}(\|u+v\|^2 - \|u\|^2 - \|v\|^2)$ in the last equality.

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

$$\sum_{t=0}^{N-1} (f(x^{t+1}) + g(y^{t+1}) - f(\bar{x}) - g(\bar{y})) \leq \frac{1}{2} \left(\frac{1}{\gamma\beta} \|z^0\|^2 + \|y_e^0\|_{\mathcal{T}}^2 \right) + \frac{\beta}{2} \sum_{t=0}^{N-1} \|\mathcal{A}x_e^{t+1}\|^2 + \frac{\gamma-1}{2\gamma^2\beta} \sum_{t=0}^{N-1} \|z^{t+1} - z^t\|^2. \quad (28)$$

To derive an upper bound for $\sum_{t=0}^{N-1} \|\mathcal{A}x_e^{t+1}\|^2$, observe from $\mathcal{S} := 2\Sigma_f - (\beta + \mu)\mathcal{A}^*\mathcal{A} \succeq \delta I$ and (20) that

$$\|x_e^{t+1}\|^2 \leq \frac{1}{\delta} \|x_e^{t+1}\|_{\mathcal{S}}^2 \leq \frac{1}{\delta} \left(\frac{1}{\gamma\beta} \|z_e^t\|^2 + \|y_e^t\|_{\mathcal{T}}^2 \right) - \frac{1}{\delta} \left(\frac{1}{\gamma\beta} \|z_e^{t+1}\|^2 + \|y_e^{t+1}\|_{\mathcal{T}}^2 \right);$$

and thus

$$\sum_{t=0}^{N-1} \|\mathcal{A}x_e^{t+1}\|^2 \leq \|\mathcal{A}\|^2 \sum_{t=0}^{N-1} \|x_e^{t+1}\|^2 \leq \frac{\|\mathcal{A}\|^2}{\delta} \left(\frac{1}{\gamma\beta} \|z_e^0\|^2 + \|y_e^0\|_{\mathcal{T}}^2 - \frac{1}{\gamma\beta} \|z_e^N\|^2 - \|y_e^N\|_{\mathcal{T}}^2 \right) \quad (29)$$

On the other hand, using the definition of σ and choosing $\alpha = \beta$ in (20), we obtain that

$$\|z^{t+1} - z^t\|^2 \leq \frac{\gamma^2\beta}{\sigma} \left(\frac{1}{\gamma\beta} \|z_e^t\|^2 + \|y_e^t\|_{\mathcal{T}}^2 \right) - \frac{\gamma^2\beta}{\sigma} \left(\frac{1}{\gamma\beta} \|z_e^{t+1}\|^2 + \|y_e^{t+1}\|_{\mathcal{T}}^2 \right),$$

which implies that

$$\sum_{t=0}^{N-1} \|z^{t+1} - z^t\|^2 \leq \frac{\gamma^2\beta}{\sigma} \left(\frac{1}{\gamma\beta} \|z_e^0\|^2 + \|y_e^0\|_{\mathcal{T}}^2 \right) - \frac{\gamma^2\beta}{\sigma} \left(\frac{1}{\gamma\beta} \|z_e^N\|^2 + \|y_e^N\|_{\mathcal{T}}^2 \right) \quad (30)$$

Combining (29) and (30) with (28) and the convexity of f and g , the upper bound in (24) follows immediately. Furthermore, (25) follows immediately from (30), the definition of z^{t+1} and the convexity of norm function. Finally, the lower bound in (24) follows from (25) and the fact that

$$f(x) + g(y) - \langle \bar{z}, \mathcal{A}x + \mathcal{B}y - c \rangle \geq f(\bar{x}) + g(\bar{y}),$$

for all x, y , since \bar{z} is a Lagrange multiplier for (13). \square

The proximal AMA is closely related to the proximal ADMM, which has been studied extensively recently in the literature; see, for example, [6, 10, 15, 41, 42, 45]. In essence, the latter algorithm is also applicable to solve (13) and is obtained by replacing the first subproblem in (16) with

$$x^{t+1} = \arg \min_x \left\{ f(x) - \langle z^t, \mathcal{A}x \rangle + \frac{\beta}{2} \|\mathcal{A}x + \mathcal{B}y^t - c\|^2 + \frac{1}{2} \|x - x^t\|_{\mathcal{S}}^2 \right\}, \quad (31)$$

for some positive semidefinite linear map \mathcal{S} . In [10, 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 $\gamma\beta$ with $\gamma \in (0, \frac{\sqrt{5}+1}{2})$.

In the case when the proximal mapping of $\frac{1}{\beta}f$ is easy to compute, the subproblem (31) 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 (16) developed in Section 3 to (9), or equivalently, (1). To this end, letting $\tau \geq \beta\|\mathcal{M}^*\mathcal{M}\|$ so that $\mathcal{T} := \tau\mathcal{I} - \beta\mathcal{M}\mathcal{M}^* \succeq 0$, we replace the second subproblem in (10) by

$$y^{t+1} = \arg \min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^*y \rangle + \frac{\beta}{2}\|x^{t+1} + \mathcal{M}^*y\|^2 + \frac{1}{2}\|y - y^t\|_{\mathcal{T}}^2 \right\}, \quad (32)$$

where $\|\cdot\|_{\mathcal{T}}$ 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 (1) (or, equivalently, the proximal AMA as applied to (9)) can be solved efficiently. From the first-order optimality condition, the first subproblem in (10) amounts to finding x^{t+1} so that $z^t \in \partial h^*(x^{t+1})$, which can be obtained by setting $x^{t+1} = \nabla h(z^t)$ by [31, Theorem 23.5]. Moreover, the second subproblem (32) can be reformulated as

$$\begin{aligned} y^{t+1} &= \arg \min_y \left\{ P^*(y) + \langle b, y \rangle - \langle z^t, \mathcal{M}^*y \rangle + \frac{\beta}{2}\|x^{t+1} + \mathcal{M}^*y\|^2 + \frac{1}{2}\|y - y^t\|_{\mathcal{T}}^2 \right\} \\ &= \arg \min_y \left\{ P^*(y) + \langle b - \mathcal{M}z^t + \beta\mathcal{M}x^{t+1} + \beta\mathcal{M}\mathcal{M}^*y^t, y \rangle + \frac{\tau}{2}\|y - y^t\|^2 \right\} \\ &= \arg \min_y \left\{ P^*(y) + \frac{\tau}{2} \left\| y - \left(y^t - \frac{b - \mathcal{M}z^t + \beta\mathcal{M}x^{t+1} + \beta\mathcal{M}\mathcal{M}^*y^t}{\tau} \right) \right\|^2 \right\} \\ &= \text{prox}_{\tau^{-1}P^*} \left(\frac{\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1}}{\tau} \right), \end{aligned}$$

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

$$\begin{aligned} y^{t+1} &= \frac{\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1}}{\tau} - \arg \min_u \left\{ P(\tau u) + \frac{\tau}{2} \left\| u - \frac{\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1}}{\tau} \right\|^2 \right\} \\ &= \tau^{-1} \left(\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1} - \arg \min_v \left\{ P(v) + \frac{1}{2\tau} \left\| v - (\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1}) \right\|^2 \right\} \right) \\ &= \tau^{-1} \left(\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1} - \text{prox}_{\tau P}(\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}x^{t+1}) \right). \end{aligned} \quad (33)$$

Thus, the second subproblem (32) can also be efficiently solved.

We note that the only difference between this new algorithm and the proximal gradient algorithm (3) (which, we recall, is equivalent to (10)) lies in the second subproblem in (10) (and also the choice of stepsizes $\beta > 0$ and $\gamma > 0$), where we simplified the subproblem by adding a proximal term. With our choice of $\mathcal{T} = \tau\mathcal{I} - \beta\mathcal{M}\mathcal{M}^*$, 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 $(y^0, z^0) \in \mathcal{Y} \times \mathcal{Z}$, $\beta \in (0, \frac{2}{L})$, $\gamma \in (0, 1 + \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\})$, $\tau \geq \beta\|\mathcal{M}^*\mathcal{M}\|$ and $\mathcal{T} = \tau\mathcal{I} - \beta\mathcal{M}\mathcal{M}^*$.

Step 1. Set

$$\begin{cases} y^{t+1} = \text{prox}_{\tau^{-1}P^*} \left(\frac{\mathcal{T}y^t - b + \mathcal{M}z^t - \beta\mathcal{M}\nabla h(z^t)}{\tau} \right), \\ z^{t+1} = z^t - \gamma\beta(\nabla h(z^t) + \mathcal{M}^*y^{t+1}). \end{cases}$$

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 (1) 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 (1).*

Proof. Recall that $\{(x^t, y^t, z^t)\}$ thus generated is the same as the sequence obtained from the proximal AMA for solving (9), using the same initial points, β , γ and \mathcal{T} .

We first show that **A1** holds for (9). To this end, let \bar{z} be a solution of (1). Since h is continuous and differentiable, from the first-order optimality condition, (2) and the subdifferential calculus rules [31, Theorem 23.8], [31, Theorem 23.9], we have

$$0 \in \nabla h(\bar{z}) + \mathcal{M}^*\partial P(\mathcal{M}\bar{z} - b).$$

Hence, there exists $\bar{y} \in \partial P(\mathcal{M}\bar{z} - b)$ and $\bar{x} := \nabla h(\bar{z})$ so that $\bar{x} + \mathcal{M}^*\bar{y} = 0$. Furthermore, notice from [31, Theorem 23.5] that

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

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

Next, recall from [26, Theorem 2.1.5] that for any $z_1, z_2 \in \mathcal{Z}$, 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 ∇h is Lipschitz continuous with modulus bounded by L . Hence h^* satisfies (14) in place of f with $\Sigma_f = \frac{1}{L}\mathcal{I}$.

The conclusion of the theorem now follows immediately from a direct application of Theorem 1, with $\Sigma_f = \frac{1}{L}\mathcal{I}$ and $\Sigma_g = 0$. \square

Remark 1. *Unlike the proximal gradient algorithm (3), the sequence $\{\mathcal{M}z^t - b\}$ generated using $\{z^t\}$ from the above algorithm does not necessarily lie in $\text{dom } P$. This could be a disadvantage since then the objective function F of (1) is not necessarily defined at each z^t . However, when $\gamma = 1$, notice from (32) that*

$$\begin{aligned} 0 &\in \partial P^*(y^{t+1}) + \underbrace{b - \mathcal{M}z^t + \beta\mathcal{M}(x^{t+1} + \mathcal{M}^*y^{t+1})}_{b - \mathcal{M}z^{t+1}} + \mathcal{T}(y^{t+1} - y^t) \\ &\Rightarrow y^{t+1} \in \partial P(\mathcal{M}z^{t+1} - b - \mathcal{T}(y^{t+1} - y^t)). \end{aligned} \tag{34}$$

Hence, in the case when $\gamma = 1$ and \mathcal{M} is surjective so that $\mathcal{T} = \mathcal{M}\mathcal{R}$ for some linear map \mathcal{R} , one can define

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

Then we see immediately from (34) that $\{\mathcal{M}\tilde{z}^t - b\} \subseteq \text{dom } P$. Furthermore, since $\{y^t\}$ is convergent by Theorem 3, it follows that $\lim_{t \rightarrow \infty} \tilde{z}^t = \lim_{t \rightarrow \infty} z^t$. Thus, the sequence $\{\tilde{z}^t\}$ is a feasible minimizing sequence for (1).

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

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

$$\min_z \frac{1}{2}\|z - \bar{z}\|^2 + P(\mathcal{M}z - b), \quad (35)$$

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

$$\begin{cases} y^{t+1} = \text{prox}_{\tau^{-1}P^*} \left(\frac{(\tau\mathcal{I} - \mathcal{M}\mathcal{M}^*)y^t + \mathcal{M}\bar{z} - b}{\tau} \right), \\ z^{t+1} = \bar{z} - \mathcal{M}^*y^{t+1}; \end{cases} \quad (36)$$

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 \mathcal{M} is surjective with $\mathcal{T} = \mathcal{M}\mathcal{R}$ for some linear map \mathcal{R} , one can obtain a feasible sequence $\{\tilde{z}^t\}$ converging to the optimal solution of (35) as

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

On the other hand, one can show that the y -update in (36) is exactly the updating rule obtained by applying the proximal gradient algorithm (3) to the dual of (35). Indeed, according to (9), the dual of (35) is given by

$$\min_y \underbrace{\frac{1}{2}\|\mathcal{M}^*y\|^2 - \langle \mathcal{M}\bar{z} - b, y \rangle}_{h_2(y)} + P^*(y). \quad (37)$$

Notice that τ is an upper bound of the Lipschitz continuity modulus of the gradient of the smooth part $h_2(y)$ of the objective function in (37). Thus, the updating rule (3) applied to (37) yields the iterates

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

which is just the y -update in (36).

We illustrate the PPG algorithm when $\mathcal{M}^*\mathcal{M} = \mathcal{I}$ or $\mathcal{M}\mathcal{M}^* = \mathcal{I}$ in the next example.

Example 2. Suppose that $\mathcal{M}^*\mathcal{M} = \mathcal{I}$ so that $\|\mathcal{M}^*\mathcal{M}\| = 1$. One can then take $\tau = \beta$ and $\gamma = 1$ in the PPG algorithm. Using the definition of z^{t+1} , (33) and introducing the auxiliary iterates u^t and w^t , one can easily verify that the iterates of the algorithm can be represented as

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

On the other hand, if $\mathcal{M}\mathcal{M}^* = \mathcal{I}$, then setting $\tau = \beta$ will give $\mathcal{T} = 0$. Hence, no proximal term is added to the alternating minimization algorithm when applied to the dual problem (9), and the PPG algorithm (with $\tau = \beta = \frac{1}{L}$ and $\gamma = 1$) for (1) 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:

$$\min_z h(z) + \sum_{i=1}^m P_i(z), \quad (38)$$

where the proximal mappings for τP_i , $\tau > 0$ and $i = 1, \dots, m$ are all easy to compute. The problem (38) is readily written in the form of (1) and hence the PPG algorithm is applicable. More precisely, define $\mathcal{M}z = (z, z, \dots, z) \in \mathcal{Z}^m$ and $P(u_1, \dots, u_m) = \sum_{i=1}^m P(u_i)$. Then we have $\sum_{i=1}^m P_i(z) = P(\mathcal{M}z)$. Moreover, the proximal mapping of τP , $\tau > 0$, 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 (1) with $\beta = \frac{1}{L}$, $\tau = \beta\|\mathcal{M}^*\mathcal{M}\|$ and $\gamma = 1$. 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

$$\begin{aligned} & -2\|\bar{z}\| \sqrt{\frac{L}{N} \sqrt{L\|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\tau\mathcal{I}-\beta\mathcal{M}\mathcal{M}^*}^2}} \leq h^*(\bar{x}^N) + P^*(\bar{y}^N) - h^*(\bar{x}) - P^*(\bar{y}) \\ & \leq \frac{1}{2N} (L\|z^0\|^2 + \|y^0 - \bar{y}\|_{\tau\mathcal{I}-\beta\mathcal{M}\mathcal{M}^*}^2) + \frac{1}{N} (L\|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\tau\mathcal{I}-\beta\mathcal{M}\mathcal{M}^*}^2). \end{aligned}$$

and

$$\|\mathcal{A}\bar{x}^N + \mathcal{B}\bar{y}^N - c\| \leq 2\sqrt{\frac{L}{N} \sqrt{L\|z^0 - \bar{z}\|^2 + \|y^0 - \bar{y}\|_{\tau\mathcal{I}-\beta\mathcal{M}\mathcal{M}^*}^2}},$$

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

Proof. Recall that the PPG algorithm in this case is just the proximal AMA applied to (9) with $\beta = \frac{1}{L}$ and $\gamma = 1$. Due to the choice of γ and β , we see that δ can be chosen to be $\frac{1}{2L}$ (and thus $\mu = \frac{1}{2L}$, where μ is defined as in Theorem 2) and σ can be chosen to be $\min\{\frac{1}{2}, \frac{\mu L}{2}\} = \frac{1}{4}$ in Theorem 2. The conclusion now immediately follows from Theorem 2. \square

Relationship with inexact proximal gradient algorithms. The PPG algorithm can be interpreted as a variant of an *inexact* proximal gradient algorithm. Recall that an inexact proximal gradient algorithm is an algorithm in which the subproblem (3) is only solved inexactly in each iteration, and convergence is guaranteed if each subproblem (3) is solved up to a certain accuracy; see, for example, [29, 33].

To describe such an interpretation, we start by presenting one approach to solve the subproblem (3) via an iterative method. In this approach, we first notice that (3) can be equivalently written

as

$$\min_z P(\mathcal{M}z - b) + \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 \quad (39)$$

$$\begin{aligned} &= \min_z \max_y \left\{ \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 + \langle y, \mathcal{M}z - b \rangle - P^*(y) \right\} \\ &\geq \max_y \min_z \left\{ \frac{L}{2} \left\| z - \left(z^t - \frac{1}{L} \nabla h(z^t) \right) \right\|^2 + \langle y, \mathcal{M}z - b \rangle - P^*(y) \right\} \\ &= \max_y \min_z -\frac{1}{2L} \|\mathcal{M}^*y\|^2 + \langle \mathcal{M}\hat{z}^t - b, y \rangle + \frac{L}{2} \left\| z - \hat{z}^t + \frac{1}{L} \mathcal{M}^*y \right\|^2 - P^*(y) \quad (40) \end{aligned}$$

$$= \max_y \underbrace{-\frac{1}{2L} \|\mathcal{M}^*y\|^2 + \langle \mathcal{M}\hat{z}^t - b, y \rangle - P^*(y)}_{-h_3(y)}, \quad (41)$$

where $\hat{z}^t := z^t - \frac{1}{L} \nabla h(z^t)$, the first equality follows from the definition of convex conjugate of P and (7). Furthermore, from (2) and [31, Theorem 31.2], we see that (39) indeed equals (41). This implies that equality holds throughout in the above relations. Thus, in order to solve for (39), one can equivalently solve for (41). Moreover, it follows from (40) that one can recover the optimal solution of (39) from an optimal solution \tilde{y}^{t+1} of (41) by setting

$$z^{t+1} = z^t - \frac{1}{L} (\nabla h(z^t) + \mathcal{M}^* \tilde{y}^{t+1}). \quad (42)$$

We thus discuss an algorithm for solving (41) instead. Recall that the proximal mapping of $\tau^{-1}P^*$ is easy to compute for $\tau > 0$, assuming that of τP , $\tau > 0$, is easy to compute; see (8) and the discussions leading to (33). Hence, the maximization problem in (41) can be efficiently solved using the proximal gradient algorithm. In an inexact proximal gradient algorithm based on this scheme, one solves (41) sufficiently accurately by the proximal gradient algorithm and updates z^{t+1} according to (42), using the approximate solution in place of \tilde{y}^{t+1} .

However, instead of getting a sufficiently accurate solution for (41), suppose we go to the other extreme: apply *one step* of the proximal gradient algorithm to (41), initialized at y^t . Then, the resulting point y^{t+1} is given by

$$y^{t+1} = \text{prox}_{\tau^{-1}P^*} \left(y^t - \frac{\beta \mathcal{M} \mathcal{M}^* y^t - \mathcal{M} z^t + \beta \mathcal{M} \nabla h(z^t) + b}{\tau} \right) \quad (43)$$

where $\beta = \frac{1}{L}$ and $\tau \geq \beta \|\mathcal{M}^* \mathcal{M}\|$, the Lipschitz continuity modulus of the gradient of h_3 in (41). Notice that the PPG algorithm with $\gamma = 1$ just consists of (43) and (42), 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. This relationship between the PPG algorithm and the proximal gradient algorithm resembles the relationship between the linearized Bregman iterative algorithm and the Bregman iterative algorithm (see, for example, [44, Section 5.3]), and that between proximal ADMM and ADMM (see, for example, [42, Section 2]).

On the other hand, this approach is fundamentally different from those considered in [33] which require the subproblem to be solved up to a certain accuracy. Hence, their convergence results do not directly apply to our algorithm. The result in [29, 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.

Relationship with an algorithm proposed in [5]. The algorithms proposed in [5] apply to a more general objective function and allow inexact computation of gradients and proximal mappings; see [39] for a further generalization of the algorithms into the settings of finding zeroes for monotone operators. When restricted to solving our problem (1) using exact gradient and proximal mapping computations and constant stepsize in extrapolation, one of their algorithms becomes quite similar to the PPG algorithm. More precisely, that algorithm in this special case reduces to

$$\begin{cases} z^{t+1} = z^t - \beta(\nabla h(z^t) + \mathcal{M}^*y^t), \\ y^{t+1} = \text{prox}_{\tau^{-1}P^*} \left(y^t - \frac{2\beta\mathcal{M}\mathcal{M}^*y^t - \mathcal{M}z^t + 2\beta\mathcal{M}\nabla h(z^t) + b}{\tau} \right), \\ (z^{t+1}, y^{t+1}) = \gamma(z^{t+1}, y^{t+1}) + (1 - \gamma)(z^t, y^t). \end{cases} \quad (44)$$

Although the formulae for updating the variables are similar to our algorithm, there are quite a number of differences. First, their order of updating the variables are different, and the extrapolation is done for *both* variables. Moreover, they use $\tau\mathcal{I} - 2\beta\mathcal{M}\mathcal{M}^*$ to scale y^t instead of $\tau\mathcal{I} - \beta\mathcal{M}\mathcal{M}^*$. Convergence result is given in [5, Theorem 3.1], where β and τ have to be chosen so that

$$\frac{1}{\beta} - \frac{\|\mathcal{M}^*\mathcal{M}\|}{\tau} \geq \frac{L}{2} \quad \text{and} \quad 0 < \gamma < 2 - \frac{L}{2} \left(\frac{1}{\beta} - \frac{\|\mathcal{M}^*\mathcal{M}\|}{\tau} \right)^{-1}.$$

Thus, while β can still be chosen from $(0, \frac{2}{L})$, their τ has to be chosen so that $\tau \geq \frac{\beta\|\mathcal{M}^*\mathcal{M}\|}{1 - \frac{\beta L}{2}}$, which can be considerably larger than our choice of τ especially when β is large. Furthermore, unlike our algorithm where the maximum possible γ depends *only* on β and L , it depends also on the choice of τ for their algorithm: the smaller the τ , the closer the maximum possible γ is to 1, meaning that the effect of extrapolation becomes smaller. In view of these differences, it does not seem that this algorithm proposed in [5] is equivalent 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 [19, Section II.B] and the fused lasso logistic regression problem [22, 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 [19, Section II.B] (see also [10, Section 5]), the system realization problem modeled as a nuclear norm minimization problem takes the following form:

$$\min_{z \in \mathbb{R}^{m \times n(j+k-1)}} p_{\text{sys}}(z) = \frac{1}{2} \|w \circ z - w \circ \hat{z}\|^2 + \lambda \|\mathcal{H}(z)\|_*, \quad (45)$$

where $z = (z_0 \ z_1 \ \cdots \ z_{j+k-2})$ with each $z_i \in \mathbb{R}^{m \times n}$, \hat{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 $i = 0, \dots, k-1$ and is zero otherwise, \circ denotes the Hadamard (entry-wise) product, $\|\cdot\|$ denotes the Fröbenius 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 $\mathcal{H}(z)$ is

a block Hankel matrix defined as

$$\mathcal{H}(z) = \begin{pmatrix} z_0 & z_1 & \cdots & z_{k-1} \\ z_1 & z_2 & \cdots & z_k \\ \vdots & \vdots & & \vdots \\ z_{j-1} & z_j & \cdots & z_{j+k-2} \end{pmatrix} \in \mathbb{R}^{mj \times nk}.$$

It is easy to see that (45) is in the form of (1) with

$$h(z) = \frac{1}{2} \|w \circ z - w \circ \hat{z}\|^2, \quad P(u) = \lambda \|u\|_*, \quad \mathcal{M} = \mathcal{H}, \quad b = 0.$$

From these we see that the condition (2) is trivially satisfied. Moreover, as mentioned in the introduction, the proximal mapping of τP , $\tau > 0$, is easy to compute. Furthermore, since $z \mapsto \|\mathcal{H}(z)\|_*$ is coercive, the set of optimal solutions of (45) is nonempty. Hence, all three assumptions on (1) are satisfied for the specific problem (45). For ease of reference, we also write down the dual of (45) as follows:

$$\begin{aligned} \min_{\nu, y} \quad & d_{\text{sys}}(\nu) := \frac{1}{2} \|\nu\|^2 + \langle w \circ \hat{z}, \nu \rangle \\ \text{s.t.} \quad & w \circ \nu + \mathcal{H}^*(y) = 0, \quad \|y\|_\infty \leq \lambda, \end{aligned} \quad (46)$$

where $\|y\|_\infty$ denotes the operator norm of $y \in \mathbb{R}^{mj \times nk}$, where ν 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 [10, Section 5.1] for solving (45). To apply our algorithm, we have to determine a bound L for the Lipschitz continuity modulus of ∇h , and obtain a bound of $\|\mathcal{M}^* \mathcal{M}\|$. It is easy to see one can take $L = 1$. Furthermore, it follows from [10, 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 [24, Section 6.2], $L_{\mathcal{M}}$ can be chosen to be $\frac{1}{2}(1 + \sqrt{1 + 4 \min\{j, k\}})$. Finally, we refer the readers to [10, Section 5.1] for details about the primal ADMM2.

We initialize all three algorithms at the origin, i.e., $(y^0, z^0) = (0, 0)$. We terminate the algorithms when

$$\max \left\{ \frac{|\min_{s \in \Gamma} p_{\text{sys}}(z^s) + d_{\text{sys}}(-w \circ \mathcal{H}^*(\text{proj}_\Omega(y^t)))|}{\max\{\min_{s \in \Gamma} p_{\text{sys}}(z^s), 1\}}, \frac{5 \|\mathcal{H}^*(\text{proj}_\Omega(y^t)) - w \circ \mathcal{H}^*(\text{proj}_\Omega(y^t))\|}{\max\{\|\mathcal{H}^*(\text{proj}_\Omega(y^t))\|, 1\}} \right\} < tol \quad (47)$$

for some $tol > 0$, where proj_Ω denotes the projection onto the set $\Omega := \{y : \|y\|_\infty \leq \lambda\}$,³ the set $\Gamma := \{s : 1 \leq s \leq t, s \text{ is a multiple of } 10\}$, and the above criterion (47) is checked every 10 iterations. In our test below, we take $tol = 1e - 4$ for all three algorithms. Also, for our PPG algorithm, we take $\tau = \beta \min\{j, k\}$, $\gamma = 1 + 0.95 \min\{0.5, \frac{1}{\beta L} - 0.5\}$, with $\beta = \frac{1}{L} = 1$ for $\lambda = 0.05$ and $\beta = 0.05$ otherwise,⁴ while for the MFBS method (6), we take $L_{\mathcal{M}} = \frac{1}{2}(1 + \sqrt{1 + 4 \min\{j, k\}})$ and $\sigma = 0.95$. We use the same parameters as used in [10, Section 5.1] for the primal ADMM2.

We generate random instances as in [19, Section II(B)]. We start by generating random matrices $A \in \mathbb{R}^{r \times r}$, $B \in \mathbb{R}^{r \times n}$ and $C \in \mathbb{R}^{n \times r}$ 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

³For the MFBS method, we used (u^t, v^t) in place of (z^t, y^t) in (47). On the other hand, the projection in (47) is only performed for the primal ADMM2.

⁴While it is intuitively obvious that for a fixed β , one should take γ as large as possible and τ as small as possible so as to maximize the stepsizes for updating z and y , respectively, the situation is not so clear for the choice of β . This is because β is directly proportional to the stepsize in the z -update but inverse proportional to the stepsize in the y -update. Thus, we experimented with $\beta = \frac{0.05}{L}$, $\frac{1}{L}$ and $\frac{1.95}{L}$. Setting $\beta = \frac{0.05}{L}$ seems to work best for $\lambda \geq 0.1$; while for $\lambda = 0.05$, setting $\beta = \frac{1}{L}$ works best.

$t = 0, \dots, T - 1$ with i.i.d. standard Gaussian entries, and create an “output” \tilde{u}_t , $t = 0, \dots, T - 1$, according to the state space model:

$$\begin{aligned} v_{t+1} &= Av_t + Be_t, \\ \tilde{u}_t &= Cv_t + e_t. \end{aligned}$$

Random noise is then added to the output \tilde{u} to give $\hat{u} = \tilde{u} + \sigma\epsilon$, with ϵ having i.i.d. standard Gaussian entries. Finally, we set, for each $i = 0, \dots, k - 1$,

$$\hat{z}_i = \frac{1}{T} \sum_{t=0}^{T-1-i} \hat{u}_{t+i} \hat{u}_t^T,$$

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 (45) at termination (**pobj**), dual objective value (46) at termination (**dobj**) and the relative dual infeasibility (**dfeas**)

$$\frac{\|\mathcal{H}^*(\text{proj}_\Omega(y^t)) - w \circ \mathcal{H}^*(\text{proj}_\Omega(y^t))\|}{\max\{\|\mathcal{H}^*(\text{proj}_\Omega(y^t))\|, 1\}}.$$

We observe that the primal ADMM2 is usually the fastest, and our PPG algorithm is usually faster than the MFBS method.

Table 1: Results for PPG algorithm, primal ADMM2 and MFBS method on solving (45)

k	λ	PPG			P. ADMM2			MFBS		
		iter	cpu	pobj/dobj/dfeas	iter	cpu	pobj/dobj/dfeas	iter	cpu	pobj/dobj/dfeas
100	0.05	123	7.4	6.073e+0/6.072e+0/4.3e-6	40	2.7	6.073e+0/6.073e+0/9.6e-6	108	7.3	6.073e+0/6.073e+0/1.5e-5
100	0.10	82	4.7	7.419e+0/7.419e+0/1.7e-5	78	5.1	7.419e+0/7.419e+0/4.2e-7	299	19.4	7.419e+0/7.419e+0/1.9e-6
100	0.50	58	3.8	1.180e+1/1.180e+1/6.3e-6	21	1.5	1.180e+1/1.180e+1/4.2e-8	97	6.9	1.180e+1/1.180e+1/4.8e-6
200	0.05	41	4.8	1.014e+1/1.014e+1/1.1e-5	40	5.2	1.014e+1/1.014e+1/5.2e-6	191	24.3	1.014e+1/1.014e+1/1.9e-5
200	0.10	100	11.7	1.288e+1/1.288e+1/1.7e-5	55	7.3	1.288e+1/1.288e+1/3.9e-6	177	22.8	1.288e+1/1.288e+1/4.9e-6
200	0.50	51	5.9	1.756e+1/1.755e+1/5.5e-6	20	2.5	1.755e+1/1.755e+1/4.2e-9	93	12.1	1.756e+1/1.755e+1/2.7e-6
300	0.05	30	5.0	1.259e+1/1.259e+1/1.4e-5	38	7.0	1.259e+1/1.259e+1/6.4e-6	224	43.2	1.259e+1/1.259e+1/1.9e-5
300	0.10	156	28.7	1.768e+1/1.768e+1/1.8e-5	43	9.1	1.768e+1/1.767e+1/1.3e-5	95	20.1	1.768e+1/1.767e+1/1.2e-5
300	0.50	53	8.8	2.253e+1/2.253e+1/3.9e-6	20	3.6	2.253e+1/2.253e+1/2.3e-9	111	20.9	2.253e+1/2.253e+1/2.0e-6

5.2 Fused lasso logistic regression problem

As discussed in [22, Section 5], the fused lasso logistic regression problem can be presented as follows:

$$\min_{z \in \mathbb{R}^n} p_{\text{flasso}}(z) := \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T \tilde{z} + z_n))) + \lambda_1 \sum_{i=1}^{n-1} |z_i| + \lambda_2 \sum_{i=1}^{n-2} |z_{i+1} - z_i|, \quad (48)$$

where $a_i \in \mathbb{R}^m$ are samples, $b_i \in \{-1, 1\}$, $i = 1, \dots, m$, $m < n$, $\lambda_1 \geq 0$ 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 (48) is in the form of (1) with

$$\begin{aligned} h(z) &= \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T \tilde{z} + z_n))), \\ P(u) &= \lambda_1 \sum_{i=1}^{n-1} |u_i| + \lambda_2 \sum_{i=n}^{2n-3} |u_i|, \quad \mathcal{M} = M := \begin{pmatrix} I_{n-1} & 0 \\ E_{n-2} & 0 \end{pmatrix}, \quad b = 0, \end{aligned}$$

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 i th row is given by $(-b_i a_i^T \quad -b_i)$ 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 (9) of (48) is given by

$$\begin{aligned} \min_{\nu, y} \quad & d_{\text{flasso}}(\nu) := \sum_{i=1}^m (\nu_i \log(\nu_i) + (1 - \nu_i) \log(1 - \nu_i)) \\ \text{s.t.} \quad & A^T \nu + M^T y = 0, \quad \max_{1 \leq i \leq n-1} |y_i| \leq \lambda_1, \quad \max_{n \leq i \leq 2n-3} |y_i| \leq \lambda_2, \end{aligned} \quad (49)$$

where A^T and M^T are the transpose of the matrices A and M , respectively, and ν is related to the x in (9) via $x = A^T \nu$. Since $m < n$, under a reasonable assumption that A has full row rank, we have $\nu = (A^T)^\dagger x$, where \dagger denotes the pseudoinverse. First, as mentioned in the introduction, the proximal mapping of τP , $\tau > 0$, is easy to compute. Moreover, since (2) is trivially satisfied for (48), there is a zero duality gap between the primal problem (48) and the dual problem (49). From the dual problem, it is not hard to observe that if b_i are not all the same and both λ_1 and λ_2 are positive, then the generalized Slater condition is satisfied for (49) and an optimal solution to (48) exists by [31, Corollary 28.2.2]. Hence, all assumptions on (1) are satisfied for the specific problem (48), under the reasonable assumptions that b_i are not all the same (meaning that there are samples from the two different classes) and both λ_1 and λ_2 are positive.

In this subsection, we will perform numerical experiments to compare our PPG algorithm against the MFBS method (6) on solving (48). To apply our algorithm, we have to upper-bound the Lipschitz continuity modulus of ∇h and $\|\mathcal{M}^* \mathcal{M}\|$. To this end, it is routine to show that ∇l is Lipschitz continuous with modulus bounded by 0.25. It then follows immediately from $h(z) = l(Az)$ that ∇h is Lipschitz continuous with modulus bounded by $0.25 \lambda_{\max}(A^T A)$, the maximum eigenvalue of $A^T A$. Next, notice that

$$M^T M = \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 & 0 \\ -1 & 3 & -1 & \cdots & \cdots & 0 & 0 \\ 0 & -1 & 3 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 3 & -1 & 0 \\ 0 & \cdots & \cdots & 0 & -1 & 2 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 3 & -1 & 0 & \cdots & \cdots & 0 & 0 \\ -1 & 3 & -1 & \cdots & \cdots & 0 & 0 \\ 0 & -1 & 3 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 3 & -1 & 0 \\ 0 & \cdots & \cdots & 0 & -1 & 3 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} =: M_2,$$

while the maximum eigenvalue of M_2 is bounded above by 5; see, for example, [16, Page 525]. Hence, we have $\|\mathcal{M}^* \mathcal{M}\| \leq 5$. On the other hand, to apply the MFBS method (6), we need to bound the Lipschitz continuity modulus of $G(z, y)$. Using [24, Lemma 6.2] and the bound $\|\mathcal{M}^* \mathcal{M}\| \leq 5$, we see that $L_{\mathcal{M}}$ can be chosen to be $\frac{1}{2}(0.25 \lambda_{\max}(A^T A) + \sqrt{(0.25 \lambda_{\max}(A^T A))^2 + 20})$.

For simplicity, we initialize both algorithms at the origin, i.e., $(y^0, z^0) = (0, 0)$. We terminate the algorithms when

$$\max \left\{ \frac{|\min_{s \in \Gamma} p_{\text{flasso}}(z^s) + d_{\text{flasso}}(\tilde{v}^t)|}{\max\{\min_{s \in \Gamma} p_{\text{flasso}}(z^s), 1\}}, \frac{5 \|A^T \tilde{v}^t + M^T y^t\|}{\max\{\|A^T \tilde{v}^t\|, \|M^T y^t\|, 1\}} \right\} < \text{tol} \quad (50)$$

for some $tol > 0$,⁵ 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 (50) is checked every 500 iterations. In our test below, we take $tol = 1e-4$ for both algorithms. Moreover, for our PPG algorithm, we take⁶ $\beta = \frac{1.95}{L} = \frac{7.8}{\lambda_{\max}(A^T A)}$, $\tau = \frac{39}{\lambda_{\max}(A^T A)}$ and $\gamma = 1 + 0.95 \min\{0.5, \frac{1}{\beta L} - 0.5\}$, while for the MFBS method (6), we take $L_{\mathcal{M}} = \frac{1}{2}(0.25\lambda_{\max}(A^T A) + \sqrt{(0.25\lambda_{\max}(A^T A))^2 + 20})$ 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 [22, Section 6.2] and [43, Section 3.1], we take

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

where $\xi_i, i = 1, \dots, 4$, are random numbers following standard Gaussian distribution. We then set b to be the sign vector of $C\hat{x} + \xi_5 e$, where ξ_5 is a random number in $[0, 1]$ and e is the vector of all ones as in [22, 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 (48) is then formed as $(C' \quad -b)$. Furthermore, we set $\lambda_1 = \alpha m$ and $\lambda_2 = 100\lambda_1$ for some $\alpha > 0$, mimicking the choice in [22, 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 (**iter**), CPU time in seconds (**cpu**), primal objective value (48) at termination (**obj**), dual objective value (49) at termination (**dobj**) and the relative dual infeasibility

$$\frac{\|A^T \tilde{v}^t + M^T y^t\|}{\max\{\|A^T \tilde{v}^t\|, \|M^T y^t\|, 1\}}$$

at termination (**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 α is smaller.

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 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.

⁵For the MFBS method, we used (u^t, v^t) in place of (z^t, y^t) in (50), and set $x^{t+1} = \nabla h(u^t)$ for all t .

⁶We experimented with $\beta = \frac{0.05}{L}, \frac{1}{L}$ and $\frac{1.95}{L}$. Setting $\beta = \frac{1.95}{L}$ seems to work best for all values of α .

Table 2: Results for PPG algorithm on solving (48)

n	α	iter	cpu	pobj/dobj/dfeas
10000	1e-4	6450	16.7	1.167e+2/1.167e+2/1.1e-5
10000	3e-4	2400	6.1	1.560e+2/1.560e+2/7.9e-6
10000	5e-4	1500	3.8	1.670e+2/1.670e+2/5.3e-6
20000	1e-4	5700	39.4	1.262e+2/1.262e+2/1.3e-5
20000	3e-4	2950	22.1	1.491e+2/1.491e+2/9.3e-6
20000	5e-4	1600	12.5	1.631e+2/1.631e+2/4.4e-6
30000	1e-4	8150	97.7	1.144e+2/1.145e+2/1.1e-5
30000	3e-4	2900	31.1	1.504e+2/1.504e+2/7.2e-6
30000	5e-4	1850	21.9	1.652e+2/1.652e+2/8.8e-6

Acknowledgements. The author would like to thank Christopher Jordan-Squire for carefully proofreading an early version of this manuscript, and Stephen Beck for pointing out the important references [5] and [39].

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] L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optimiz. Theory App.* 158, pp. 460–479 (2013).
- [6] J. Eckstein. Some saddle-function splitting methods for convex programming. *Optim. Method Softw.* 4, pp. 75–83 (1994).
- [7] 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).
- [8] M. Elad, P. Milanfar and R. Rubinstein. Analysis versus synthesis in signal priors *Inverse Probl.* 23, pp. 947–968 (2007).
- [9] 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).
- [10] M. Fazel, T. K. Pong, D. Sun and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. A.* 34, pp. 946–977 (2013).
- [11] 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).

- [12] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, eds., *Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems*. North-Holland, Amsterdam, 1983.
- [13] 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).
- [14] 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 Op’erATIONnelle*. 9 (R-2), pp. 41–76 (1975).
- [15] 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).
- [16] N. J. Higham. *Accuracy and Stability of Numerical Algorithms* SIAM, Philadelphia, 2nd edition (2002).
- [17] 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).
- [18] 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).
- [19] 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).
- [20] Z. Liu, A. Hansson and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Syst. Control Lett.* 62, pp. 605–612 (2013).
- [21] S. Ma, D. Goldfarb and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* 128, pp. 321–353 (2011).
- [22] 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>.
- [23] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. *Proc. American Control Conference* (2010).
- [24] 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).
- [25] 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).
- [26] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers (2003).
- [27] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.* 16, pp. 235–249 (2005).
- [28] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.* 103, pp. 127–152 (2005).
- [29] M. Patriksson. Cost approximation: a unified framework of descent algorithms for nonlinear programs. *SIAM J. Optim.* 8, pp. 561–582 (1998).

- [30] 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).
- [31] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton (1970).
- [32] 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).
- [33] M. W. Schmidt, N. Le Roux and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS*, pp. 1458–1466 (2011).
- [34] 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).
- [35] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.* 29, pp. 119–138 (1991).
- [36] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.* 38, pp. 431–446 (2000).
- [37] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Program.* 125, pp. 263–295 (2010).
- [38] L. Vandenberghe. Convex optimization techniques in system identification. *Proc. IFAC Symposium on System Identification*, pp. 71–76 (2012).
- [39] B. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.* 38, pp. 667–681 (2013).
- [40] H. Wang and A. Banerjee. Online alternating direction method. Preprint, June 2013. Available at <http://arxiv.org/abs/1306.3721>.
- [41] M. H. Xu and T. Wu. A class of linearized proximal alternating direction methods. *J. Optim. Theory Appl.* 151, pp. 321–337 (2011).
- [42] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1 -problems in compressive sensing. *SIAM J. Sci. Comput.* 33, pp. 250–278 (2011).
- [43] G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Comput. Stat. Data An.* 55, pp. 1552–1569 (2011).
- [44] W. Yin, S. Osher, J. Darbon and D. Goldfarb. Bregman iterative algorithms for compressed sensing and related problems. *SIAM J. Imaging Sci.* 1, pp. 143–168 (2008).
- [45] 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).