A Primal-Dual Algorithmic Framework for Constrained Convex Minimization

Quoc Tran-Dinh • Volkan Cevher

Laboratory for Information and Inference Systems (LIONS),
École Polytechnique Fédérale de Lausanne (EPFL), CH1015 - Lausanne, Switzerland.
{quoc.trandinh, volkan.cevher}@epfl.ch.

June 20, 2014

Abstract

We present a primal-dual algorithmic framework to obtain approximate solutions to a prototypical constrained convex optimization problem, and rigorously characterize how common structural assumptions affect the numerical efficiency. Our main analysis technique provides a fresh perspective on Nesterov's excessive gap technique in a structured fashion and unifies it with smoothing and primal-dual methods. For instance, through the choices of a dual smoothing strategy and a center point, our framework subsumes decomposition algorithms, augmented Lagrangian as well as the alternating direction method-of-multipliers methods as its special cases, and provides optimal convergence rates on the primal objective residual as well as the primal feasibility gap of the iterates for all.

Keywords: Primal-dual method; optimal first-order method; augmented Lagrangian; alternating direction method of multipliers; separable convex minimization; monotropic programming; parallel and distributed algorithm.

1 Introduction

This article is concerned about the following constrained convex minimization problem, which captures a surprisingly broad set of problems in various disciplines [11, 18, 43, 69]:

$$f^\star := \min_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathscr{X}\}, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed and convex function; $\mathscr{X} \subseteq \mathbb{R}^n$ is a nonempty, closed and convex set; and $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ are known. In the sequel, we develop efficient numerical methods to approximate an optimal solution $\mathbf{x}^\star$ to (1) and rigorously characterize how common structural assumptions on (1) affect the efficiency of the methods.

1.1 Scalable numerical methods for (1) and their limitations

In principle, we can obtain high accuracy solutions to (1) through an equivalent unconstrained problem [13, 54]. For instance, when $\mathscr{X}$ is absent and f is smooth, we can eliminate the linear constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$ by using a projection onto the null-space of $\mathbf{A}$ and then applying well-understood smooth minimization techniques. Whenever available, we can also exploit barrier representations of the constraints $\mathscr{X}$ and avoid non-smooth f via reformulations, such as *lifting*, as in the interior point method using disciplined convex programming

[13, 33, 46, 48]. While the resulting smooth and unconstrained problems are simpler than (1) in theory, the numerical efficiency of the overall strategy severely suffers from the curse-of-dimensionality as well as the loss of the numerical structures in the original formulation.

Alternatively, we can obtain low- or medium-accuracy solutions when we augment the objective $f(\mathbf{x})$ with simple penalty functions on the constraints. For instance, we can solve

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + (\rho/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \ : \ \mathbf{x} \in \mathscr{X} \right\}, \tag{2}$$

where $\rho > 0$ is a penalty parameter. Despite the fundamental difficulties in choosing the penalty parameter, this approach enhances our computational capabilities as well as numerical robustness since we can apply modern proximal gradient, alternating direction, and primal-dual methods. Intriguingly, the scalability of virtually all these solution algorithms rely on three key structures that stand out among many others:

Structure 1 (Decomposability): We say that the constrained problem (1) is *p-decomposable* if the objective function f and the feasible set $\mathscr{X}$ can be represented as follows

$$f(\mathbf{x}) := \sum_{i=1}^{p} f_i(\mathbf{x}_i), \ \text{ and } \ \mathscr{X} := \prod_{i=1}^{p} \mathscr{X}_i, \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$, $\mathscr{X}_i \in \mathbb{R}^{n_i}$, $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$ is proper, closed and convex for $i = 1, \ldots, p$, and $\sum_{i=1}^{p} n_i = n$. Decomposability immediately supports parallel and distributed implementations in synchronous hardware architectures. This structure arises naturally in linear programming, network optimization, multi-stages models and distributed systems [11]. With decomposability, the problem (1) is also referred to as a monotropic convex program [63].

Structure 2 (Proximal tractability): Unconstrained problems can still pose significant difficulties in numerical optimization when they include non-smooth terms. However, many non-smooth problems (e.g., of the form (2)) can be solved nearly as efficiently as smooth problems, provided that the computation of the proximal operator is **tractable**[1] [4, 58, 62]:

$$\text{prox}_{\lambda f}(\mathbf{x}) := \arg\min_{\mathbf{z} \in \mathscr{X}} \left\{ f(\mathbf{z}) + (1/(2\lambda))\|\mathbf{z}-\mathbf{x}\|_2^2 \right\}, \tag{4}$$

where $\lambda > 0$ is a constant. While the proximal operators simply use $\mathscr{X} = \mathbb{R}^n$ in the canonical setting, we employ (4) to do away with the $\mathscr{X}$-feasibility of the algorithmic iterates. Many smooth and non-smooth functions support efficient proximal operators [18, 21, 43, 69]. Clearly, decomposability proves useful in the computation of (4).

Structure 3 (Special function classes): Often times, the function f in (1) or the individual terms f_i in (3) possess additional properties that can enhance numerical efficiency. Table 1 highlights common properties that are typically (but not necessarily) associated with function smoothness. These structures provide iterative algorithms with analytic upper and lower bounds on the objective (or its gradient), and aid the theoretical design of their iterations as well as their practical step-size and momentum parameter selection [4, 13, 48, 54, 66].

On the basis of these structures, we can design algorithms featuring a full spectrum of (nearly) dimension-independent, global convergence rates for composite convex minimization problems with well-understood analytical complexities [4, 48, 53, 52, 66]. Unfortunately, the scalable, penalty-based approaches above invariably feature one or both of the following two drawbacks which blocks their full impact.

[1] It can be solved in a closed form, low computational cost or polynomial time.

Table 1: Special convex function classes. In the optimization literature, we refer to L, σ, and ν as the Lipschitz, strong convexity, and barrier parameters, respectively.

Class	Name	Property $\mathbf{x},\mathbf{y} \in \mathrm{dom}\,(f)\,,\ \mathbf{v} \in \mathbb{R}^n, 0 \leq \sigma \leq L < +\infty$
$\mathscr{F}_L$	Lipschitz gradient	$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x}-\mathbf{y}\|$
$\mathscr{F}_\sigma$	Strong convexity	$\frac{\sigma}{2}\|\mathbf{x}-\mathbf{y}\|^2 + f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \leq f(\mathbf{y})$
$\mathscr{F}^2$	Standard self-concordant	$\lvert\varphi'''(t)\rvert \leq 2\varphi''(t)^{3/2}$: $\varphi(t) = f(\mathbf{x}+t\mathbf{v})$, $t \in \mathbb{R}$
$\mathscr{F}^{2,\nu}$	Self-concordant barrier	$\mathscr{F}^2$ and $\sup_{\mathbf{v}\in\mathbb{R}^n}\left\{2\nabla f(\mathbf{x})^T\mathbf{v} - \|\mathbf{v}\|_{\mathbf{x}}^2\right\} \leq \nu$

Limitation 1 (Non-ideal convergence characterizations): Ideally, the convergence characterization of an algorithm for solving (1) must establish rates both on absolute value of the primal objective residual $\left|f(\mathbf{x}^k) - f^\star\right|$ as well as the primal feasibility of its linear constraints $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|$, simultaneously on its iterates $\mathbf{x}^k \in \mathscr{X}$. The constraint feasibility is critical so that the primal convergence rate has any significance. Rates on weighted primal objective residual and feasibility gap is not necessarily meaningful since (1) is a constrained problem and $f(\mathbf{x}^k) - f^\star$ can easily be negative at all times as compared to the unconstrained setting where we trivially have $f(\mathbf{x}^k) - f^\star \geq 0$.

Table 2 demonstrates that the convergence results for some existing methods are far from ideal. Most algorithms have guarantees in the ergodic sense (i.e., on the averaged history of iterates without any weight) [15, 37, 38, 57, 64, 70] with non-optimal rates, which diminishes the practical performance; they rely on special function properties to improve convergence rates on the function and feasibility [56, 57], which reduces the scope of their applicability; they provide rates on dual functions [32], or a weighted primal residual and feasibility score [64], which does not necessarily imply convergence on the absolute value of the primal residual or the feasibility; or they obtain convergence rate on the gap function value sequence composed both the primal and dual variables via variational inequality and gap function characterizations [15, 37, 38], where the rate is scaled by a diameter parameter which is not necessary bounded.[2]

Limitation 2 (Computational inflexibility): Recent theoretical developments customize algorithms to exploit special function classes for scalability. We have indeed moved away from the black-box model of optimization, which forms the foundation of the interior point method's flexibility, where, for instance, we restrict ourselves to compute solely the values and the (sub)gradients of the objective and the constraints at a point.

Unfortunately, specialized algorithms requires knowledge of function class parameters, do not address the full scope of (1) (e.g., with self-concordant functions or fully non-smooth decompositions), and often have complicated algorithmic implementations with backtracking steps, which create computational bottlenecks. Moreover, these issues are further compounded by their penalty parameter selection, such as ρ in (2) (cf., [12] for an extended discussion), which can significantly decrease numerical efficiency, as well as the inability to handle p-decomposability in an optimal fashion, which rules out parallel architectures for their computation.

1.2 Our contributions

To this end, we address the following two questions in this paper: "Is it possible to efficiently solve (1) using only the proximal tractability assumption with global convergence guarantees?" and "Can we actually charac-

[2]We refer to the standard ADMM (see, e.g., [12]) and not the parallel ADMM variant or multi-block ADMM, which can have convergence guarantees given additional assumptions.

Table 2: Illustrative convergence guarantees for solving (1) under the proximal tractability assumption. Note that most convergence rate results in the table are in the ergodic or *averaged* sense, where $\widehat{\mathbf{x}}^k = k^{-1}\sum_{i=1}^k \mathbf{x}^i$.

Method name	Assumptions	Convergence	References
ADMM	$\leq$ 2-decomposable	$\mathscr{O}(1/k)$ on the joint $(\mathbf{x}^k, \mathbf{y}^k)$ using a gap function	[15, 37, 38]
Decomposition method	p-decomposable	$f(\widehat{\mathbf{x}}^k) - f^\star + r\|\mathbf{A}\widehat{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k)$ $(r > 0)$	[64]
[Fast] ADMM	$\leq$ 2-decomposable and f_1 or $f_2 \in \mathscr{F}_\mu$	$[\mathscr{O}(1/k^2)]$ $\mathscr{O}(1/k)$ on the dual-objective	[32]
Bregman ADMM	$\leq$ 2-decomposable	$f(\widehat{\mathbf{x}}^k) - f^\star \leq \mathscr{O}(1/k)$ and $\|\mathbf{A}\widehat{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/\sqrt{k})$	[70]
Fast Linearized ADMM	$\leq$ 2-decomposable and f_1 or $f_2 \in \mathscr{F}_L$	$f(\widehat{\mathbf{x}}^k) - f^\star \leq \mathscr{O}(1/k)$ and $\|\mathbf{A}\widehat{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k)$	[57]
Primal-Dual Hybrid Gradient (PDHG)	Saddle point problem	$\mathscr{O}(1/k)$ based on gap function values composed both primal-dual variables	[31]
[Inexact] augmented Lagrangian method	$\leq$ 2-decomposable	$\left\|f(\mathbf{x}^k) - f^\star\right\| \leq \mathscr{O}(1/k^2)$ and $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k^2)$ (*non-ergodic*)	*This work*
Decomposition methods [Inexact] 1P2D and 2P1D	p-decomposable	$\left\|f(\mathbf{x}^k) - f^\star\right\| \leq \mathscr{O}(1/k)$ and $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k)$ (*non-ergodic*)	*This work*
	p-decomposable and $f_i \in \mathscr{F}_\sigma$	$\left\|f(\mathbf{x}^k) - f^\star\right\| \leq \mathscr{O}(1/k^2)$, $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k^2)$, and $\|\mathbf{x}^k - \mathbf{x}^\star\|_2 \leq \mathscr{O}(1/k)$ (*non-ergodic*)	
New ADMM and its preconditioned variants	$\leq$ 2-decomposable	$\left\|f(\mathbf{x}^k) - f^\star\right\| \leq \mathscr{O}(1/k)$ and $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k)$ (*non-ergodic*)	*This work*

terize the convergence rate of the primal objective residual and primal feasibility gap separately?" The answer is indeed positive provided that there exists a solution in a bounded primal feasible set $\mathscr{X}$.

Surprisingly, we can still exploit favorable function classes, such as $\mathscr{F}_L$ and $\mathscr{F}_\sigma$ when available, optimally exploit p-decomposability and its special 2-decomposable sub-case, and have a penalty parameter-free black-box optimization method. The second question is also important since in primal-dual framework, trade-off between the primal objective residual and the primal feasibility gap is crucial, which makes algorithm numerically stable, see, e.g., [31] for numerical examples.

To achieve the desiderata, we unify primal-dual methods [10, 61], smoothing [50, 61], and the excessive gap function technique introduced in [49] in convex optimization.

Primal-dual methods: Primal-dual methods rely on strong duality in convex optimization [60] and are also related to many other methods for solving saddle points, monotone inclusions and variational inequalities [28]. In our approach, we reformulate the optimality condition of (1) as a mixed-variational inequality and use the gap function as our main tool to develop the algorithms.

Smoothing: Smoothing techniques are widely used in optimization to replace non-smooth functions with differentiable approximations. In this work, we describe two smoothing strategies for the dual function of (1) in the Lagrange formulation based on Bregman distances and the augmented Lagrangian technique. We show that the augmented Lagrangian smoother preserves convergence properties for the algorithm to solve (1) and feature a convergence rate independent of the spectral norm of $\mathbf{A}$. In addition, the Bregman smoother allows us to handle p-decomposability by only relying on the proximal tractability assumption.

Excessive gap function: Excessive gap technique was introduced by Nesterov in [49] and has been used to develop primal-dual solution methods for solving nonsmooth unconstrained problems. In this paper, we exploit the same excessive gap idea but in a structured form for a variational inequality characterizing the optimality condition of (1). We then combine these three existing techniques in order to develop a unified primal-dual framework for solving (1) and analyze the convergence of its algorithmic instances under mild assumptions.

Our specific theoretical and practical contributions are as follows:

i) We present a unified primal-dual framework for solving constrained convex optimization problems of the form (1). This framework covers augmented Lagrangian method [39, 45], (preconditioned) ADMM [15], proximal-based decomposition [20] and decomposition method [67] as special cases, which we make explicit in Section 6.

ii) We prove the convergence and establish rates for three variants (cf., Theorem 4.1) of our algorithmic framework without any need to select a penalty parameter. An important result is the convergence rate in a non-ergodic sense of both primal objective residual $\left|f(\bar{\mathbf{x}}^k) - f^\star\right| \leq \mathcal{O}(1/k^\alpha)$ and the primal feasibility gap $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq \mathcal{O}(1/k^\alpha)$, where $\alpha = 1$ or 2. Our rates are considered optimal given our particular assumptions (cf., Table 2).

iii) We consider an inexact variant of our algorithmic framework for the special case of 2-decomposability, which allows one to solve the subproblems up to given predetermined accuracy so that it still maintains the same worst-case analytical complexity as in the exact case provided that the accuracy of solving the subproblems is controlled appropriately. This variant allows us to handle 2-decomposability with only proximal tractability assumption.

iv) We show how special function classes can be exploited and describe their convergence implications.

Our characterization is radically different from existing results such as in [5, 15, 23, 37, 38, 57, 64]. We clarify the importance of this result in Section 4 as well as Section 6 in the context of existing convergence results for ADMM and its variants. For the p-decomposability, the variants corresponding to our Bregman smoothing technique can be implemented in a fully parallel and distributed manner, where the feasibility guarantee acts as a consensus rate. In special case, where $p = 2$, we propose a strategy to enhance the practical convergence rate by trading off the objective residual with the feasibility gap.

On the computational front, we test our algorithms on several well-studied numerical problems using both synthetic and real-world data, compare them to other existing state-of-the-art methods, and provide open-source code for each application. We also discuss the update of the smoothness parameters in order to enhance the performance of the algorithms by trading-off between the optimality gap and the feasibility gap. Numerical results show the advantages of our methods on several numerical tests.

1.3 Related work

Due to the generality of (1), there has been an explosion of interest in the convex optimization in developing solution algorithms for it. Unfortunately, it is impossible to provide a comprehensive summary of the ever-expanding literature in any reasonable space. Hence, this subsection attempts to relate some important algorithmic frameworks for solving (1) to our work with selected, representative citations in each.

Methods-of-multipliers/primal-dual methods: One of the oldest primal-dual methods for solving (1) is the method-of-multipliers (MoM), which is based on Lagrange dualization [10]. Without further assumptions on

f and $\mathscr{X}$, the dual step of this method can be viewed as a subgradient iteration, which features a provably slow convergence rate, i.e., $\mathscr{O}(1/\sqrt{k})$, where k is the iteration count. MoM is also known to be sensitive to the step-size selection rules for damping the search direction.

In order to overcome the difficulty of nonsmoothness in the dual function, several attempts have been made. For instance, we can add either a proximal term or an augmented term to the Lagrange function of (1) to smooth the dual function [20, 34, 35, 44, 45, 61]. Intriguingly, while the specific methods studied in [20, 34, 35, 61] are quite borad, no global convergence rate has been established so far.

The works in [44, 45] provide convergence rates by applying Nesterov's accelerated scheme to the dual problem of (1). In recent paper [64], the authors shows that the method proposed in [20] has convergence rate $\mathscr{O}(1/k)$. However, this convergence rate is a joint between the objective residual and the primal feasibility gap, i.e., $f(\mathbf{x}^k) - f^\star + r\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2 \leq \mathscr{O}(1/k)$ for $r > 0$ given. We note that this convergence rate on the weighted measure does not imply the convergence rate of $\left|f(\mathbf{x}^k) - f^\star\right|$ and $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2$ separately in constrained optimization.

In [27] the author studies several variants of the primal-dual algorithm and presented several applications in image processing. Convergence analysis of these variants are also presented in [27], however the global convergence rate has not been provided. In [31], the authors describe a primal-dual hybrid gradient (PDHG) algorithm, which can be considered as a variant of the same primal-dual algorithm. In [31], the authors also studied several heuristic strategies to update the parameters, and show that the convergence rate of this algorithm is $\mathscr{O}(1/k)$ in an ergodic sense with respect to a VIP gap function values.

Methods from monotone inclusions and variational inequalities: The optimality condition of (1) can be viewed as a monotone inclusion or a mixed variational inequality (VIP) corresponding to both the primal and dual variables $[\mathbf{x}, \mathbf{y}] \in \mathscr{X} \times \mathbb{R}^m$. As a result, we can leverage algorithms from these two respective fields to solve (1) [15, 28, 37, 38]. For instance, the work in [15] exploit the idea from variational inequality proposed in [47, 51]. Splitting methods including Douglas-Rachford and predictor-corrector methods considered [21, 22, 26, 36, 55] also belong to this direction. However, since monotone inclusions or variational inequalities are much more general than (1), using methods from these fields typically lead to inefficient algorithms in practice for solving the specific optimization problem (1).

Augmented Lagrangian and alternating direction methods: Augmented Lagrangian (AL) methods have come to offer an important computational perspective on a broad class of constrained convex problems of the form (1). In this setting, we first define the Lagrangian function associated with the linear constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$ of (1) as $\mathscr{L}(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$. Then, we introduce the augmented Lagrangian function: $\mathscr{L}_\gamma(\mathbf{x}, \mathbf{y}) := \mathscr{L}(\mathbf{x}, \mathbf{y}) + (\gamma/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ for a given penalty parameter $\gamma > 0$. Classical augmented Lagrangian method [11] solving (1) produces a sequence $\left\{(\mathbf{x}^k, \mathbf{y}^k)\right\}_{k\geq 0}$ starting from $(\mathbf{x}^0, \mathbf{y}^0) \in \mathscr{X} \times \mathbb{R}^m$ as

$$\left\{\begin{array}{ll} \mathbf{x}^{k+1} & := \arg\min_{\mathbf{x}\in\mathscr{X}} \mathscr{L}_\gamma(\mathbf{x}, \mathbf{y}^k), \\ \mathbf{y}^{k+1} & := \mathbf{y}^k + \gamma(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}), \end{array}\right. \tag{5}$$

Under a suitable choice of γ, it is well-known that method (5) converges to a global optimal $(\mathbf{x}^\star, \mathbf{y}^\star)$ of (1) at $\mathscr{O}(1/k)$ rate under mild assumptions, i.e., $\mathscr{L}(\mathbf{x}^k, \mathbf{y}^k) - \mathscr{L}(\mathbf{x}^\star, \mathbf{y}^\star) \leq \mathscr{O}(1/k)$. In fact, this method can be accelerated by applying Nesterov's accelerating scheme [48] to obtain $\mathscr{O}(1/k^2)$ convergence rate.

Within the class of augmented Lagrangian methods, perhaps the most famous variant is the alternating direction method of multipliers (ADMM), which appears in many guises in the literature. This method has been recognized as a special case of Douglas-Rachford splitting algorithm applying to its optimality condition [12, 26, 32]. In ADMM, given that f and $\mathscr{X}$ are separable with $p = 2$. This case also covers the composite minimization problem of the form $\min_{\mathbf{x}_1\in\mathbb{R}^{n_1}} f_1(\mathbf{x}_1) + f_2(\mathbf{A}\mathbf{x}_1)$, where both f_1 and f_2 are convex. By using a slack variable, we can reformulate the composite problem into (1) as $\min_{\mathbf{x}\in\mathbb{R}^n} f_1(\mathbf{x}_1) + h(\mathbf{x}_2)$ subject to

$\mathbf{A}\mathbf{x}_1 = \mathbf{x}_2$. In the ADMM context, the first problem in (5) can be solved iteratively as

$$\left\{\begin{array}{ll} \mathbf{x}_1^{k+1} & := \arg\min\limits_{\mathbf{x}_1 \in \mathscr{X}_1} \Big\{ f_1(\mathbf{x}_1) + (\mathbf{y}^k)^T \mathbf{A}_1 \mathbf{x}_1 + (\gamma/2)\|\mathbf{A}_1\mathbf{x}_1 - \mathbf{x}_2^k\|_2^2 \Big\}, \\ \mathbf{x}_2^{k+1} & := \arg\min\limits_{\mathbf{x}_2 \in \mathscr{X}_2} \Big\{ f_2(\mathbf{x}_2) + (\mathbf{y}^k)^T \mathbf{A}_2 \mathbf{x}_2 + (\gamma/2)\|\mathbf{A}_1\mathbf{x}_1^{k+1} - \mathbf{x}_2\|_2^2 \Big\}. \end{array}\right. \tag{6}$$

The main computational difficulty of ADMM is the $\mathbf{x}_1$-update problem (i.e., the first subproblem) in (6). Indeed, we have to numerically solve this step in general except when $\mathbf{A}^T\mathbf{A}$ is efficiently diagonalizable. Interestingly, the diagonalization step in many cases can be done via Fourier Transform. Many notable applications support this feature, such as matrix completion where $\mathbf{A}$ models sub-sampled matrix entries, image deblurring where $\mathbf{A}$ is a convolution operator, and total variation regularization where $\mathbf{A}$ is a differential operator with periodic boundary conditions. We can also circumvent this computational difficulty by using a preconditioned ADMM variant [15].

ADMM is one of the most popular method in practice. However, its efficiency depends significantly on the choice of the penalty parameter γ. Unfortunately, theoretical guarantee for choosing this parameter is still an open problem and is not yet well-understood. When f_1 is strongly convex, we can drop the quadratic term in the first line of (6) in order to obtain an alternating minimization algorithm (AMA) [68]. This method turns out to be a forward-backward splitting algorithm for its optimality inclusion [32].

A note on [50]: We note that the approach presented in this paper builds upon the excessive gap idea in [50]. Technically, we use the same idea but in a much structured fashion, whereby we enforce a particular linear form in preserving the excessive gap as shown in Definition 3.2. This particular structure is key in obtaining our rates.

Moreover, there are several fundamental differences between our algorithmic framework and the methods studied in [49] as a result of the excessive gap technique. First, we use augmented Lagrangian functions and Bregman distances for smoothing the dual problem of (1). Second, we consider the Lagrangian primal-dual formulation for (1) where we do not have the boundedness of the feasible set of the dual variable. In this case the key estimate [50, estimate (3.3)] does not apply to our setting. Third, we update all algorithmic parameters simultaneously and do not need an odd-even switching strategy [49, Method 1: b) and c)] Four, we do not assume that the objective function f of (1) has Lipschitz gradient which is required in [49]. This are several important applications, where this assumption simply does not hold [43]. Fifth, our method is applied to the constrained problem (1), which requires the feasibility gap characterization as opposed to unconstrained problems where we only need to worry about the optimality.

1.4 Paper organization

The rest of this paper is organized as follows. In the next section, we recall basic concepts, and introduce a mixed-variational inequality formulation of (1). In Section 3, we propose two key smoothing techniques for (1), called the Bregman and augmented Lagrangian smoothing techniques. We also provide a formal definition for the excessive gap function from [50] and further investigate its properties. Section 4 presents the main primal-dual algorithmic framework for solving (1) and its convergence theory. Section 5 specifies different instances of our algorithmic framework for (1) under given assumptions. Section 6 makes further connections to existing methods in the literature. Section 7 is devoted to implementation issues and Section 8 presents numerical simulations. The appendix provides detail proofs of the theoretical results in the main text.

2 Preliminaries

First we recall the well-known definition of the Bregman distance, the primal-dual formulation for (1), and a variational inequality characterization for the optimality condition of (1), which will be used in the sequel.

2.1 Basic notation

Given a proper, closed and convex function f, we denote $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}$ the domain of f, $\partial f(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^n \mid f(\tilde{\mathbf{x}}) - f(\mathbf{x}) \geq \mathbf{v}^T(\tilde{\mathbf{x}} - \mathbf{x}),\ \forall \tilde{\mathbf{x}} \in \mathrm{dom}(f)\}$ the subdifferential of f at $\mathbf{x}$. If f is differentiable, $\nabla f(\mathbf{x})$ denotes the gradient of f at $\mathbf{x}$. For given vector $\mathbf{x} \in \mathbb{R}^n$, we define $\|\mathbf{x}\|_2$ the Euclidean norm of $\mathbf{x}$. We use a superscripted notation $L^f > 0$ to denote the corresponding Lipschitz constant of a differentiable function f. Similarly, we use a subscripted notation $\sigma_g > 0$ to denote the corresponding strong convexity constant of a convex function g.

2.2 Proximity functions and Bregman distances

Given a nonempty, closed convex set $\mathscr{X}$, a nonnegative, continuous and σ_b-strongly convex function b is called a *proximity* function (or prox-function) of $\mathscr{X}$ if $\mathscr{X} \subseteq \mathrm{dom}(b)$. For example, the simplest prox-function is $b_{\mathscr{X}}(\mathbf{x}) := (\sigma_b/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$ for any $\sigma_b > 0$ and $\mathbf{x}_c \in \mathscr{X}$. Whenever unspecified, we use this specific prox-function with $\sigma_b = 1$.

Given a smooth prox-function b of $\mathscr{X}$ with the parameter $\sigma_b > 0$. We define

$$d_b(\mathbf{x}, \mathbf{y}) := b(\mathbf{x}) - b(\mathbf{y}) - \nabla b(\mathbf{y})^T(\mathbf{x} - \mathbf{y}),\ \ \forall \mathbf{x}, \mathbf{y} \in \mathrm{dom}(b), \tag{7}$$

the Bregman distance between $\mathbf{x}$ and $\mathbf{y}$ with respect to b. Given a matrix $\mathbf{S}$, we also define the projected prox-diameter of a given set $\mathscr{X}$ with respect to d_b as

$$D_{\mathscr{X}}^{\mathbf{S}} := \sup_{\mathbf{x}, \mathbf{x}_c \in \mathscr{X}} d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c). \tag{8}$$

Here, we project the set $\mathscr{X}$ onto the range space of matrix $\mathbf{S}$. If $\mathscr{X}$ is bounded, then $0 \leq D_{\mathscr{X}}^{\mathbf{S}} < +\infty$. For $b(\mathbf{x}) := (\sigma_b/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$, we have $d_b(\mathbf{x}, \mathbf{y}) = (\sigma_b/2)\|\mathbf{x} - \mathbf{y}\|_2^2$, which is indeed the Euclidean distance.

2.3 Primal-dual formulation

We write the min-max formulation of (1) based on the Lagrange dualization as follows:

$$\max_{\mathbf{y} \in \mathbb{R}^m} \min_{\mathbf{x} \in \mathscr{X}} \mathscr{L}(\mathbf{x}, \mathbf{y}) \equiv \max_{\mathbf{y} \in \mathbb{R}^m} \min_{\mathbf{x} \in \mathscr{X}} \{f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}, \tag{9}$$

where $\mathscr{L}$ is the Lagrange function and $\mathbf{y}$ is the dual variable. We write the dual function $g(\mathbf{y})$ as

$$g(\mathbf{y}) := \min_{\mathbf{x} \in \mathscr{X}} \{f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}, \tag{10}$$

which leads to the following definition of the so-called dual problem

$$g^\star := \max_{\mathbf{y} \in \mathbb{R}^m} g(\mathbf{y}). \tag{11}$$

Let $\mathbf{x}^\star(\mathbf{y})$ be a solution of (10) at a given $\mathbf{y} \in \mathbb{R}^m$. Corresponding to $\mathbf{x}^\star(\mathbf{y})$, we also define the domain of g as

$$\mathrm{dom}(g) := \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{x}^\star(\mathbf{y}) \text{ exists}\}. \tag{12}$$

If f is continuous on $\mathscr{X}$ and if $\mathscr{X}$ is compact, then $\mathbf{x}^\star(\mathbf{y})$ exists for any $\mathbf{y} \in \mathbb{R}^m$. Unfortunately, the dual function g is typically nonsmooth, and hence the numerical solutions of (11) are usually difficult [48]. In general, we have $g(\mathbf{y}) \leq f(\mathbf{x})$, which is known as weak-duality in convex optimization. In order to guarantee strong duality, i.e., $f^\star = g^\star$ for (1) and (11), we require the following assumption:

Assumption A. 1 *The constraint set $\mathscr{X}$ and the solution set $\mathscr{X}^\star$ of* (1) *are nonempty. The function f is proper, closed and convex. In addition, either $\mathscr{X}$ is a polytope or the following Slater condition holds:*

$$\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}\} \cap \operatorname{relint}(\mathscr{X}) \neq \emptyset, \tag{13}$$

where $\operatorname{relint}(\mathscr{X})$ *is the relative interior of $\mathscr{X}$.*

Under Assumption 1, the solution set $\mathscr{Y}^\star$ of the dual problem (11) is also nonempty and bounded. Moreover, the strong duality holds, i.e., $f^\star = g^\star$. Any point $(\mathbf{x}^\star, \mathbf{y}^\star) \in \mathscr{X}^\star \times \mathscr{Y}^\star$ is a primal-dual solution to (1) and (11), and is also a saddle point of the Lagrange function $\mathscr{L}$, i.e., $\mathscr{L}(\mathbf{x}^\star, \mathbf{y}) \leq \mathscr{L}(\mathbf{x}^\star, \mathbf{y}^\star) \leq \mathscr{L}(\mathbf{x}, \mathbf{y}^\star)$ for all $\mathbf{x} \in \mathscr{X}$ and $\mathbf{y} \in \mathbb{R}^m$. These inequalities lead to the following estimate

$$f(\mathbf{x}) - g(\mathbf{y}) \geq f(\mathbf{x}) - f^\star \geq -\|\mathbf{y}^\star\|_2 \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2, \ \ \forall \mathbf{x} \in \mathscr{X}, \ \mathbf{y} \in \mathbb{R}^m. \tag{14}$$

Our goal in this paper is to solve the primal constrained problem (1), while numerical algorithms only give an approximate solution up to a certain accuracy. Hence, we need to specify the concept of an approximate solution for (1).

Definition 2.1 *Given a target accuracy $\varepsilon \geq 0$, a point $\tilde{\mathbf{x}}^\star \in \mathscr{X}$ is said to be an ε-solution of* (1) *if $|f(\tilde{\mathbf{x}}^\star) - f^\star| \leq \varepsilon$ and $\|\mathbf{A}\tilde{\mathbf{x}}^\star - \mathbf{b}\|_2 \leq \varepsilon$.*

Here, we assume in Definition 2.1 that $\tilde{\mathbf{x}}^\star \in \mathscr{X}$, i.e., $\tilde{\mathbf{x}}^\star$ is exactly feasible to $\mathscr{X}$. This requirement is reasonable in practice since $\mathscr{X}$ is usually a "simple" set where the projection onto $\mathscr{X}$ can be computed exactly. Moreover, we can use different accuracy levels for the absolute value of the primal objective residual $|f(\tilde{\mathbf{x}}^\star) - f^\star|$ and the primal feasibility gap $\|\mathbf{A}\tilde{\mathbf{x}}^\star - \mathbf{b}\|_2$ in Definition 2.1.

2.4 Mixed-variational inequality formulation and gap function

Let $\mathbf{w} := (\mathbf{x}, \mathbf{y}) \equiv (\mathbf{x}^T, \mathbf{y}^T)^T \in \mathbb{R}^n \times \mathbb{R}^m$ be the primal-dual variable and $F(\mathbf{w}) := \begin{pmatrix} \mathbf{A}^T\mathbf{y} \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{pmatrix}$ be a partial Karush-Kuhn-Tucker mapping. Then, the optimality condition of (1) becomes

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + F(\mathbf{w}^\star)^T(\mathbf{w} - \mathbf{w}^\star) \geq 0, \ \ \forall \mathbf{w} \in \mathscr{X} \times \mathbb{R}^m, \tag{15}$$

which is known as a *mixed-variational inequality* [28]. If we define

$$G(\mathbf{w}^\star) := \max_{\mathbf{w} \in \mathscr{W} := \mathscr{X} \times \mathbb{R}^m} \left\{ f(\mathbf{x}^\star) - f(\mathbf{x}) + F(\mathbf{w}^\star)^T(\mathbf{w}^\star - \mathbf{w}) \right\}, \tag{16}$$

then G is known as the Auslender gap function of (15) [1].

Let $\mathscr{W} := \mathscr{X} \times \mathbb{R}^m$. Then, by the definition of F, we can see that

$$G(\mathbf{w}^\star) = \max_{(\mathbf{x}, \mathbf{y}) \in \mathscr{W}} \left\{ f(\mathbf{x}^\star) + \mathbf{y}^T(\mathbf{A}\mathbf{x}^\star - \mathbf{b}) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\mathbf{y}^\star \right\} = f(\mathbf{x}^\star) - g(\mathbf{y}^\star) \geq 0.$$

It is clear that $G(\mathbf{w}^\star) = 0$ if and only if $\mathbf{w}^\star := (\mathbf{x}^\star, \mathbf{y}^\star) \in \mathscr{W}^\star := \mathscr{X}^\star \times \mathscr{Y}^\star$, which is indeed the *strong duality* property.

3 Primal-dual smoothing techniques

This section shows how to use augmented Lagrangian functions and Bregman distances as a principled smoothing technique [48, 3] within our primal-dual framework. We can then obtain different algorithmic variants by simply choosing an appropriate prox-center at each iteration.

3.1 Dual function is a smoothable function

The dual function g defined by (10) is convex but in general nonsmooth. We approximate this function by a smoothed function g_γ defined as:

$$g_\gamma(\mathbf{y}) := \min_{\mathbf{x}\in\mathscr{X}} \left\{ f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + \gamma d_b(\mathbf{S}\mathbf{x},\mathbf{S}\mathbf{x}_c) \right\}, \tag{17}$$

where d_b is a given Bregman distance with the strong convexity parameter $\sigma_d > 0$, $\mathbf{x}_c \in \mathscr{X}$ is the prox-center of d_b, $\mathbf{S}$ is a given consistent projection matrix and $\gamma > 0$ is a [primal] *smoothness* parameter. The following definition characterizes approximation properties of the smoothed function g_γ.

Definition 3.1 ([3]) *The dual function g defined by* (10) *is called a $(\gamma, D, \bar{L}^g)$-smoothable function if there exist positive numbers γ, D and $\bar{L}^g$ and a concave and smooth function $g_\gamma : \mathrm{dom}\,(g) \to \mathbb{R}\cup\{+\infty\}$ so that:*

$$g_\gamma(\mathbf{y}) - \gamma D \le g(\mathbf{y}) \le g_\gamma(\mathbf{y}), \ \ \forall \mathbf{y} \in \mathrm{dom}\,(g). \tag{18}$$

In addition, $\nabla g_\gamma(\cdot)$ is Lipschitz continuous with a Lipschitz constant $L^g_\gamma := \gamma^{-1}\bar{L}^g$. □

We call g_γ the $(\gamma, D, \bar{L}^g)$-smoothed function of g or simply the smoothed function of g when these parameters are specified. We note that g_γ defined by (17) is not necessarily Lipschitz gradient for an arbitrary choice of $\mathbf{S}$ and $\mathbf{x}_c$. We consider two cases as follows.

3.1.1 Smoothing via augmented Lagrangian

Let us choose $d_b(\mathbf{u},\mathbf{u}_c) := (1/2)\|\mathbf{u}-\mathbf{u}_c\|_2^2$, $\mathbf{S} \equiv \mathbf{A}$ and $\mathbf{x}_c \in \mathscr{X}$ so that $\mathbf{A}\mathbf{x}_c = \mathbf{b}$. Then, we have trivially $d_b(\mathbf{S}\mathbf{x},\mathbf{S}\mathbf{x}_c) := (1/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$. As a result, the function g_γ defined by (17) becomes the augmented dual function, that is

$$\tilde{g}_\gamma(\mathbf{y}) := \min_{\mathbf{x}\in\mathscr{X}} \left\{ f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + (\gamma/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \right\}. \tag{19}$$

Here, $\mathscr{L}_\gamma(\mathbf{x},\mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + (\gamma/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$ is exactly the augmented Lagrangian of (1) associated with the linear constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$. We denote by $\tilde{\mathbf{x}}^\star_\gamma(\mathbf{y})$ the solution of (19) and $\mathrm{dom}\left(\tilde{g}_\gamma\right) := \left\{\mathbf{y}\in\mathbb{R}^m \mid \tilde{\mathbf{x}}^\star_\gamma(\mathbf{y}) \text{ exists}\right\}$. It is well-known that $\tilde{g}_\gamma$ is concave as well as smooth, and its gradient is Lipschitz continuous with a Lipschitz constant $L^{\tilde{g}}_\gamma := \gamma^{-1}$. We refer to $\tilde{g}_\gamma$ as an augmented Lagrangian smoother (in short, *AL smoother*) of g. The following lemma shows that $\tilde{g}_\gamma$ is a smoothed function of g, whose proof can be found, e.g., in [10].

Lemma 3.1 *For any $\gamma > 0$, $\tilde{g}_\gamma$ defined by* (19) *is concave and smooth. Its gradient is given by $\nabla\tilde{g}_\gamma(\mathbf{y}) = \mathbf{A}\tilde{\mathbf{x}}^\star_\gamma(\mathbf{y}) - \mathbf{b}$ and satisfies:*

$$\|\nabla\tilde{g}_\gamma(\mathbf{y}) - \nabla\tilde{g}_\gamma(\hat{\mathbf{y}})\|_2 \le L^{\tilde{g}}_\gamma\|\mathbf{y}-\hat{\mathbf{y}}\|_2, \ \ \forall \mathbf{y},\hat{\mathbf{y}} \in \mathrm{dom}\left(\tilde{g}_\gamma\right), \tag{20}$$

where $L^{\tilde{g}}_\gamma := \gamma^{-1} > 0$.

Consequently, $\tilde{g}_\gamma$ is a $(\gamma, D^{\mathbf{A}}_{\mathscr{X}}, \bar{L}^{\tilde{g}})$-smoothed function of g in the sense of Definition 3.1, i.e., $\tilde{g}_\gamma(\mathbf{y}) - \gamma D^{\mathbf{A}}_{\mathscr{X}} \le g(\mathbf{y}) \le \tilde{g}_\gamma(\mathbf{y})$, where $D^{\mathbf{A}}_{\mathscr{X}} := (1/2)\sup_{\mathbf{x}\in\mathscr{X}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$ and $\bar{L}^{\tilde{g}} := 1$.

3.1.2 Smoothing via Bregman distances

If we choose $\mathbf{S} := \mathbb{I}$ to be the identity matrix of $\mathbb{R}^n$, then the smoothed function g_γ defined by (17) becomes

$$\hat{g}_\gamma(\mathbf{y}) := \min_{\mathbf{x}\in\mathscr{X}} \left\{ f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + \gamma d_b(\mathbf{x},\mathbf{x}_c) \right\}. \tag{21}$$

Let us denote by $\hat{\mathbf{x}}^\star_\gamma(\mathbf{y})$ the solution of (21), which always exists. We refer to $\hat{g}_\gamma$ as a Bregman distance smoother (shortly, *BD smoother*) of g. The following lemma summarizes the properties of $\hat{g}_\gamma$ (see, e.g., [50, 67]):

Lemma 3.2 *The function $\hat{g}_\gamma$ defined by* (21) *satisfies:*

$$\hat{g}_\gamma(\mathbf{y}) - \gamma D^{\mathbb{I}}_{\mathscr{X}} \leq \hat{g}_\gamma(\mathbf{y}) - \gamma d_b(\mathbf{x}^\star(\mathbf{y}), \mathbf{x}_c) \leq g(\mathbf{y}) \leq g_\gamma(\mathbf{y}),\ \forall \mathbf{y} \in \mathbb{R}^m, \tag{22}$$

where $D^{\mathbb{I}}_{\mathscr{X}}$ is the prox-diameter of $\mathscr{X}$ with respect to d_b and $\mathbf{x}^\star(\mathbf{y})$ is the solution of (10)*.*

Moreover, $\hat{g}_\gamma$ is concave and smooth. Its gradient is given by $\nabla \hat{g}_\gamma(\mathbf{y}) := \mathbf{A}\hat{\mathbf{x}}^\star_\gamma(\mathbf{y}) - \mathbf{b}$ for all $\mathbf{y} \in \mathbb{R}^m$, and satisfies

$$\left\|\nabla \hat{g}_\gamma(\mathbf{y}) - \nabla \hat{g}_\gamma(\hat{\mathbf{y}})\right\|_2 \leq L^{\hat{g}}_\gamma \|\mathbf{y} - \hat{\mathbf{y}}\|_2,\ \ \forall \mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^m, \tag{23}$$

for $L^{\hat{g}}_\gamma := \frac{\|\mathbf{A}\|_2^2}{\gamma \sigma_d}$. Consequently, $\hat{g}_\gamma$ is a $(\gamma, D^{\mathbb{I}}_{\mathscr{X}}, \bar{L}^{\hat{g}})$-smoothed function of g, where $\bar{L}^{\hat{g}} := \frac{\|\mathbf{A}\|_2^2}{\sigma_d}$ and σ_d is the strong convexity parameter of d_b.

We note that if $\mathscr{X}$ is bounded and f is continuous (or $\mathscr{X} \subset \operatorname{relint}(\operatorname{dom}(f))$), then $\mathbf{x}^\star(\mathbf{y})$ always exists for any $\mathbf{y} \in \mathbb{R}^m$. In this case, the prox-diameter $D^{\mathbb{I}}_{\mathscr{X}}$ of $\mathscr{X}$ is finite. Consequently, (22) holds for all $\mathbf{y} \in \mathbb{R}^m$.

3.2 Smoothed gap function

As we observe from the previous section, the optimality condition of (1) can be represented as a variational inequality of the form (15). By using Auslender's gap function $G(\cdot)$ defined by (16), we can show that $\mathbf{w}^\star \in \mathscr{W}^\star$ is a primal-dual solution to (1) and (11). Since the gap function $G(\cdot)$ is generally nonsmooth, we smooth it by adding the following smoothing function:

$$d_{\gamma\beta}(\mathbf{w}) \equiv d_{\gamma\beta}(\mathbf{x}, \mathbf{y}) := \gamma d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (\beta/2)\|\mathbf{y}\|_2^2, \tag{24}$$

where d_b is a given Bregman distance, $\mathbf{S}$ is a projection matrix and γ and β are two positive *smoothness* parameters.

Remark 3.1 *For simplicity of our analysis, we use a simple quadratic prox-function $(\beta/2)\|\mathbf{y}\|_2^2$ in* (24) *for the dual variable $\mathbf{y}$. However, we can replace this term by $\beta d_{b_\mathbf{y}}(\mathbf{y}, \mathbf{y}_c)$, where $d_{b_\mathbf{y}}$ is a given Bregman distance and $\mathbf{y}_c$ is a given point in $\mathbb{R}^m$. However, depending on the choice of $d_{b_\mathbf{y}}$, the dual variable $\mathbf{y}^*_\beta(\cdot)$ may no longer have a closed form expression. However, the overall practical performance may be improved.*

The smoothed gap function for G is then defined as follows:

$$G_{\gamma\beta}(\bar{\mathbf{w}}) := \max_{\mathbf{w} \in \mathscr{X} \times \mathbb{R}^m} \left\{ f(\bar{\mathbf{x}}) - f(\mathbf{x}) + F(\bar{\mathbf{w}})^T(\bar{\mathbf{w}} - \mathbf{w}) - d_{\gamma\beta}(\mathbf{w}) \right\}, \tag{25}$$

where F is defined in (15). The function $G_{\gamma\beta}$ can be considered as Fukushima's gap function [29] for the variational inequality problem (15). We can see that $G_{\gamma\beta}(\bar{\mathbf{w}}) \to G_{00}(\bar{\mathbf{w}}) \equiv G(\bar{\mathbf{w}})$ as γ and $\beta \to 0^+$ simultaneously.

It is clear that the maximization problem (25) is a convex optimization problem. We denote by $\mathbf{w}^\star_{\gamma\beta}(\bar{\mathbf{w}}) := (\mathbf{x}^\star_\gamma(\bar{\mathbf{y}}), \mathbf{y}^\star_\beta(\bar{\mathbf{x}}))$ the solution of this problem. Then, by using the optimality condition of (25) we can easily check that $\mathbf{x}^\star_\gamma(\bar{\mathbf{y}})$ is the optimal solution to (17) at $\mathbf{y} := \bar{\mathbf{y}}$, while $\mathbf{y}^\star_\beta(\bar{\mathbf{x}})$ can be computed explicitly as

$$\mathbf{y}^\star_\beta(\bar{\mathbf{x}}) := \beta^{-1}(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}). \tag{26}$$

Our goal is to generate two sequences $\{\bar{\mathbf{w}}^k\}_{k\geq 0} \subseteq \mathscr{W}$ and $\{(\gamma_k, \beta_k)\}_{k\geq 0} \in \mathbb{R}^2_{++}$ so that $\{G_{\gamma_k\beta_k}(\bar{\mathbf{w}}^k)\}_{k\geq 0}$ becomes firmly contractive. We formally encode this idea using the following definition.

Definition 3.2 (Model-based Excessive Gap) *Given $\bar{\mathbf{w}}^k := (\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) \in \mathscr{W}$ and $(\gamma_k, \beta_k) > 0$, a new point $\bar{\mathbf{w}}^{k+1} := (\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1}) \in \mathscr{W}$ and $(\gamma_{k+1}, \beta_{k+1}) > 0$ so that $\gamma_{k+1}\beta_{k+1} < \gamma_k\beta_k$ is said to be firmly contractive (w.r.t. $G_{\gamma\beta}$ defined by* (25)*) if:*

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1 - \tau_k)G_k(\bar{\mathbf{w}}^k) - \psi_k, \tag{27}$$

where $G_k(\cdot) := G_{\gamma_k\beta_k}(\cdot)$, $\tau_k \in [0, 1)$ and $\psi_k \in \mathbb{R}$ are two given parameters. □

Here, the parameter τ_k and the decay term ψ_k will be specified accordantly with different algorithmic schemes.

In the context of excessive gap technique introduced by Nesterov, the smoothed gap function $G_{\mu_1\mu_2}(\bar{\mathbf{w}})$ measures the excessive gap $f_{\mu_2}(\bar{\mathbf{x}}) - \phi_{\mu_1}(\bar{\mathbf{y}})$ in [49, cf., (2.5) and (2.9)]). Hence, we will call $G_{\gamma\beta}(\bar{\mathbf{w}})$ Nesterov's smoothed gap function customized for the constrained convex problem (1). We note that the excessive gap condition $f_{\mu_2}(\bar{\mathbf{x}}) \leq \phi_{\mu_1}(\bar{\mathbf{y}})$ in [49, (3.2)] only requires $G_{\mu_1\mu_2}(\bar{\mathbf{w}}) \leq 0$. In our case, we structure this condition using the basic model in (27) so that we can manipulate τ_k and the new parameter ψ_k simultaneously to analyze the convergence of our algorithms.

In the sequel, we often assume that the second parameter ψ_k is nonnegative, which allows us to estimate the convergence rate of $\{G_k(\bar{\mathbf{w}}^k)\}_{k\geq 0}$. However, the following remark shows that the sequence $\{G_k(\bar{\mathbf{w}}^k)\}_{k\geq 0}$ can still converge to 0^+ even if ψ_k is positive. However, we find the ensuing convergence analysis to be difficult.

Remark 3.2 *Let $\{\tau_k\}_{k\geq 0} \subseteq (0,1)$ and $\{\psi_k\}_{k\geq 0}$ be sequences in Definition 3.2. If*

$$\lim_{k\to\infty} \tau_k = 0, \;\; \sum_{k=0}^{\infty} \tau_k = +\infty, \;\; \textit{and} \;\; \sum_{k=0}^{\infty} \psi_k < +\infty, \tag{28}$$

then the sequence $\{G_k(\bar{\mathbf{w}}^k)\}_{k\geq 0}$ converges to 0^+. □

From Definition 3.2, if $\{\bar{\mathbf{w}}^k\}_{k\geq 0} \subseteq \mathscr{W}$ and $\{(\gamma_k,\beta_k)\}_{k\geq 0} \in \mathbb{R}^2_{++}$ satisfy the condition (27), then we have $G_k(\bar{\mathbf{w}}^k) \leq \omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k$ by induction, where

$$\omega_k := \prod_{j=0}^{k-1}(1-\tau_j) \;\; \text{and} \;\; \Psi_k := \Psi_0 + \sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)\psi_j. \tag{29}$$

Consequently, the rate of convergence of $\{G_k(\bar{\mathbf{w}}^k)\}_{k\geq 0}$ depends on the rate of $\{\tau_k\}_{k\geq 0}$ and $\{\psi_k\}_{k\geq 0}$.

The next lemma shows the relation between problem (1) and its smoothed function g_γ and g. The proof of this lemma can be found in the appendix.

Lemma 3.3 *Let g_γ be defined by* (17) *and $G_{\gamma\beta}$ defined by* (25)*. Also, let $\{\bar{\mathbf{w}}^k\}_{k\geq 0} \subset \mathscr{W}$ and $\{(\gamma_k,\beta_k)\}_{k\geq 0} \in \mathbb{R}^2_{++}$ be the sequences satisfying Definition 3.2. Then we have*

$$f(\bar{\mathbf{x}}^k) - g_{\gamma_k}(\bar{\mathbf{y}}^k) \leq \omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k - (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2. \tag{30}$$

In addition, we also have the following bound:

$$-\|\mathbf{y}^\star\|_2\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq f(\bar{\mathbf{x}}^k) - f^\star \leq f(\bar{\mathbf{x}}^k) - g(\bar{\mathbf{y}}^k) \leq S_k, \tag{31}$$

$$\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \beta_k\Big[\|\mathbf{y}^\star\|_2 + \sqrt{\|\mathbf{y}^\star\|_2^2 + 2\beta_k^{-1}S_k}\Big] \tag{32}$$

where $S_k := \omega_k G_0(\bar{\mathbf{w}}^0) + \gamma_k D_{\mathscr{X}}^{\mathbf{S}} - \Psi_k$, provided that $\beta_k\|\mathbf{y}^\star\|_2^2 + 2S_k \geq 0$.

From Lemma 3.3 we can see that if $G_0(\bar{\mathbf{w}}^0) \leq \Psi_k$, then the primal objective residual $\big|f(\bar{\mathbf{x}}^k) - f^\star\big|$ and the primal feasibility gap $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$ of (1) are bounded by

$$\left\{\begin{array}{ll} \big|f(\bar{\mathbf{x}}^k) - f^\star\big| \leq \max\Big\{\gamma_k D_{\mathscr{X}}^{\mathbf{S}}, \big[2\beta_k D_{\mathscr{Y}^\star} + \sqrt{2\gamma_k\beta_k D_{\mathscr{X}}^{\mathbf{S}}}\big]D_{\mathscr{Y}^\star}\Big\}, \\ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq 2\beta_k D_{\mathscr{Y}^\star} + \sqrt{2\gamma_k\beta_k D_{\mathscr{X}}^{\mathbf{S}}}, \end{array}\right. \tag{33}$$

where $D_{\mathscr{Y}^\star} := \min\{\|\mathbf{y}^\star\|_2 \mid \mathbf{y}^\star \in \mathscr{Y}^\star\}$, which is the norm of a minimum norm dual solution. The estimate (33) hints that we can derive algorithms based on $\{(\gamma_k,\beta_k)\}$ whose convergence rate depends directly on how we update the sequence $\{(\gamma_k,\beta_k)\}_{k\geq 0}$.

4 The main algorithmic framework

The key objective in this section is to design a primal-dual update template from $\bar{\mathbf{w}}^k \in \mathscr{W}$ and $(\gamma_k, \beta_k) \in \mathbb{R}^2_{++}$ to $\bar{\mathbf{w}}^{k+1} \in \mathscr{W}$ and $(\gamma_{k+1}, \beta_{k+1}) \in \mathbb{R}^2_{++}$ so that the conditions in Definition 3.2 hold. We develop two distinct schemes to update $\bar{\mathbf{w}}^k$ and (γ_k, β_k) in the following two subsections.

4.1 An iteration scheme with two primal steps

Since the objective function is not necessary smooth, we consider the following mapping under Assumption 1:

$$\mathrm{prox}_{\mathbf{S}f}(\hat{\mathbf{x}}, \hat{\mathbf{y}}; \beta) := \arg\min_{\mathbf{x} \in \mathscr{X}} \left\{ f(\mathbf{x}) + \hat{\mathbf{y}}^T \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) + (\bar{L}^g/(2\beta)) \|\mathbf{S}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 \right\}, \tag{34}$$

where $\beta > 0$ and $\mathbf{S}$ is a projection matrix that satisfies the following condition:

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \leq \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2^2 + 2(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b})^T \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) + \bar{L}^g \|\mathbf{S}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2, \ \ \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathscr{X}. \tag{35}$$

An obvious choice of $\mathbf{S}$ is either $\mathbf{S} \equiv \mathbf{A}$ and $\bar{L}^g = 1$ or $\mathbf{S} \equiv \mathbb{I}$ and $\bar{L}^g = \|\mathbf{A}\|_2^2$. Since $\mathbf{A}$ is known, both are feasible. Alternatively, local variable metrics can be used here, which might lead to different adaptation and computation tradeoffs in optimization.

Now, given $\bar{\mathbf{w}}^k := (\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) \in \mathscr{W}$ and $(\gamma_k, \beta_k) \in \mathbb{R}^2_{++}$, we compute $\mathbf{x}^\star_\gamma(\bar{\mathbf{y}}^k)$ the solution of the minimization problem in (17) and $\mathbf{y}^\star_\beta(\bar{\mathbf{x}}^k)$ by (26). Then, we update the point $\bar{\mathbf{w}}^{k+1} := (\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ and $(\gamma_{k+1}, \beta_{k+1})$ based on the following scheme:

$$\left\{ \begin{array}{ll} \hat{\mathbf{x}}^k & := (1 - \tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\bar{\mathbf{y}}^k), \\ \hat{\mathbf{y}}^k & := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}), \\ \bar{\mathbf{x}}^{k+1} & := \mathrm{prox}_{\mathbf{S}f}(\hat{\mathbf{x}}^k, \hat{\mathbf{y}}^k; \beta_{k+1}), \\ \bar{\mathbf{y}}^{k+1} & := (1 - \tau_k)\bar{\mathbf{y}}^k + \tau_k \hat{\mathbf{y}}^k, \end{array} \right. \tag{2P1D}$$

where $\tau_k \in (0, 1]$ and $(\beta_{k+1}, \gamma_{k+1})$ is updated as

$$\beta_{k+1} = (1 - \tau_k)\beta_k \ \text{ and } \ \gamma_{k+1} = (1 - c_k\tau_k)\gamma_k, \tag{36}$$

for some $c_k \in (-1, 1]$, which will be specified later. It is important to note that if f is nonsmooth, solving problem (34) requires the same cost as solving (17). Therefore, we can refer to (2P1D) as a *primal-dual scheme with two primal steps*.

Remark 4.1 *If f is L_f-Lipschitz gradient, then we can replace $f(\mathbf{x})$ in the proximal step at the third line of* (2P1D) *by its linearization, which leads to the following gradient step:*

$$\mathrm{grad}_{\mathbf{S}f}(\hat{\mathbf{x}}, \hat{\mathbf{y}}; \beta) := \arg\min_{\mathbf{x} \in \mathscr{X}} \left\{ (\nabla f(\hat{\mathbf{x}}) + \mathbf{A}^T \mathbf{y})^T (\mathbf{x} - \hat{\mathbf{x}}) + (L_f/2)\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + (2\beta)^{-1} \|\mathbf{S}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 \right\}.$$

In particular, when f is p-decomposable as in (3) *and if f_i is Lipschitz gradient for some $i = 1, \ldots, p$, then we can use the gradient step for such a f_i [67].*

The following lemma provides conditions such that $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ updated by (2P1D) satisfies Definition 3.2, whose proof is deferred to the appendix.

Lemma 4.1 *Let $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ and $(\gamma_{k+1}, \beta_{k+1})$ be updated as* (2P1D) *and* (36)*. If $\mathbf{S}$ satisfies* (35) *and τ_k is chosen such that*

$$\beta_{k+1}\gamma_{k+1} \geq \bar{L}^g \tau_k^2, \tag{37}$$

then $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1}) \in \mathscr{W}$ and satisfies Definition 3.2, i.e., $G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1 - \tau_k)G_k(\bar{\mathbf{w}}^k) - \psi_k$ for $\psi_k := \frac{\tau_k^2}{2\beta_{k+1}} \|\mathbf{A}\mathbf{x}^\star_{\gamma_k}(\bar{\mathbf{y}}^k) - \mathbf{b}\|_2^2 \geq 0$.

4.2 An iteration scheme with two dual steps

Alternatively to (2P1D), we can switch from two primal steps to two dual steps. In this case, the new point $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ is updated as follows:

$$\left\{\begin{array}{ll} \hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k \mathbf{y}^\star_{\beta_k}(\bar{\mathbf{x}}^k), \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_{k+1}}(\hat{\mathbf{y}}^k), \\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + \frac{\gamma_{k+1}}{\bar{L}^g}\big(\mathbf{A}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) - \mathbf{b}\big), \end{array}\right. \tag{1P2D}$$

where $\tau_k \in (0,1)$ and the parameters β_{k+1} and γ_{k+1} are updated as (36). We refer to (1P2D) as a *primal-dual scheme with two dual steps*.

The following lemma shows that $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ updated by (1P2D) maintains (27), whose proof can also be found in the appendix.

Lemma 4.2 *Let $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ and $(\gamma_{k+1}, \beta_{k+1})$ be updated by* (1P2D) *and* (36)*, respectively. If τ_k is chosen such that*

$$\beta_{k+1}\gamma_{k+1} \geq \bar{L}^g \tau_k^2, \tag{38}$$

then $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1}) \in \mathscr{W}$ and satisfies $G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - \psi_k$ for

$$\psi_k := \tau_k(1-\tau_k)\gamma_k\big[d_b(\mathbf{S}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\mathbf{y}}^k), \mathbf{S}\mathbf{x}_c) - c_k d_b(\mathbf{S}\mathbf{x}^\star_{\gamma_{k+1}}(\bar{\mathbf{y}}^k), \mathbf{S}\mathbf{x}_c)\big] \geq 0.$$

4.3 Finding a starting point

In principle, we can start our algorithm at any point $(\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0) \in \mathscr{W}$. However, we can find a point $\bar{\mathbf{w}}^0 := (\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0) \in \mathscr{W}$ such that $G_{\gamma_0\beta_0}(\bar{\mathbf{w}}^0) \leq 0$. The following lemma shows how to compute such a point, whose proof can be found in the appendix.

Lemma 4.3 *Given $\mathbf{x}_c^0 \in \mathscr{X}$, the point $\bar{\mathbf{w}}^0 := (\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0) \in \mathscr{W}$ computed by*

$$\left\{\begin{array}{l} \bar{\mathbf{x}}^0 = \mathbf{x}^\star_{\gamma_0}(0^m), \\ \bar{\mathbf{y}}^0 := \beta_0^{-1}\left(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\right). \end{array}\right. \tag{39}$$

satisfies $G_{\gamma_0\beta_0}(\bar{\mathbf{w}}^0) \leq -\gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c) \leq 0$ provided that $\beta_0\gamma_0 \geq \bar{L}^g$.

Alternatively, the point $\bar{\mathbf{w}}^0 := (\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0) \in \mathscr{W}$ generated by

$$\left\{\begin{array}{ll} \bar{\mathbf{y}}^0 & := \beta_0^{-1}\left(\mathbf{A}\mathbf{x}_c - \mathbf{b}\right), \\ \bar{\mathbf{x}}^0 & := \mathrm{prox}_{\mathbf{S}f}(\mathbf{x}_c, \bar{\mathbf{y}}^0; \beta_0), \end{array}\right. \tag{40}$$

also satisfies $G_{\gamma_0\beta_0}(\bar{\mathbf{w}}^0) \leq -\gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c) \leq 0$ provided that $\beta_0\gamma_0 \geq \bar{L}^g$.

4.4 Updating step-size parameter

It remains to derive an update rule for the step-size τ_k in both scheme (2P1D) and (1P2D). The update rule is derived by using the same condition in both Lemma 4.1 and Lemma 4.2.

Since τ_k satisfies $\beta_{k+1}\gamma_{k+1} \geq \bar{L}^g\tau_k^2$, τ_{k+1} also satisfies the same condition, i.e., $\beta_{k+2}\gamma_{k+2} \geq \bar{L}^g\tau_{k+1}^2$. In addition, by (36), we have $\beta_{k+2} := (1-\tau_{k+1})\beta_{k+1}$ and $\gamma_{k+2} := (1-c_{k+1}\tau_{k+1})\gamma_{k+1}$. These conditions lead to $\tau_{k+1}^2 \leq (1-\tau_{k+1})(1-c_{k+1}\tau_{k+1})\tau_k^2$. Since we want to maximize the value of τ_{k+1}, we take the equality, i.e., $\tau_{k+1}^2 = (1-\tau_{k+1})(1-c_{k+1}\tau_{k+1})\tau_k^2$. The last condition leads to

$$a_{k+1} := \big(1 + c_{k+1} + \sqrt{4a_k^2 + (1-c_{k+1})^2}\big)/2, \;\text{ and }\; \tau_k := a_k^{-1}. \tag{41}$$

In addition, from Lemma 4.3, we have $\beta_0\gamma_0 \geq \bar{L}^g$. Let us choose $\beta_0 := \gamma_0^{-1}\bar{L}_g$. We need to choose $\tau_0 \in (0,1]$ such that $\gamma_1\beta_1 = (1-\tau_0)(1-c_0\tau_0)\beta_0\gamma_0 \geq \bar{L}^g\tau_0^2$. Therefore, we get

$$a_0 := \left(1 + c_0 + \sqrt{4(1-c_0) + (1+c_0)^2}\right)/2, \text{ and } \tau_0 := a_0^{-1}. \tag{42}$$

The following Lemma shows the convergence rate of a_k, β_k and $\beta_k\gamma_k$. The proof of this lemma can be found in the appendix.

Lemma 4.4 *Let $s_k := \sum_{i=1}^k c_i$. Then, the sequence $\{a_k\}$ updated by* (41) *with a_0 given by* (42) *satisfies*

$$(k + a_0 + s_k)/2 \leq a_k \leq k + a_0. \tag{43}$$

Consequently, the sequences $\{\beta_k\}$ and $\{\gamma_k\}$ updated by (36) *satisfy*

$$\frac{\bar{L}^g}{(k+a_0)^2} \leq \gamma_{k+1}\beta_{k+1} \leq \frac{4\bar{L}^g}{(k+a_0+s_k)^2}, \tag{44}$$

where $\bar{L}^g$ is given in Definition 3.1. Moreover, we also have

$$\begin{cases} \frac{\beta_0}{(k+2)^2} \leq \beta_{k+1} \leq \frac{4\beta_0}{(k+1)^2}, & \text{if } c_k = 0, \\ \beta_{k+1} = \frac{\beta_0}{k+2}, & \text{if } c_k = 1. \end{cases} \tag{45}$$

4.5 A primal-dual algorithmic template

Now, we combine all ingredients presented in the previous subsection to obtain the template for solving (1) shown in Algorithm 1.

Algorithm 1: (*Primal-dual template using model-based excessive gap technique*)

Inputs: $\gamma_0 > 0$, $c_0 \in (-1, 1]$, and a smoother (AL or BD).

Initialization:

1: $a_0 := \left(1 + c_0 + [4(1-c_0) + (1+c_0)^2]^{1/2}\right)/2$ and $\tau_0 := a_0^{-1}$.

2: Use $\bar{L}^g := 1$ for AL smoother and $\bar{L}^g := \sigma_d^{-1}\|\mathbf{A}\|_2^2$ for BD smoother.

3: $\beta_0 := \bar{L}^g/\gamma_0$.

4: Compute $(\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0)$ by either (39) or (40).

For $k = 0$ **to** $k_{\max}$

5: If **stopping_criterion**, terminate.

6: Given $(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)$, update $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ by either (2P1D) or (1P2D).

7: $\beta_{k+1} := (1-\tau_k)\beta_k$ and update $\gamma_{k+1} := (1 - c_k\tau_k)\gamma_k$.

8: Update c_{k+1} from c_k if necessary.

9: Update $a_{k+1} := \left(1 + c_{k+1} + [4a_k^2 + (1-c_{k+1})^2]^{1/2}\right)/2$ and set $\tau_{k+1} := a_{k+1}^{-1}$.

End For

The main step of Algorithm 1 is Step 5, where we need to update $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ based on either (2P1D) or (1P2D). If we use (2P1D), then γ_k can be updated as $\gamma_{k+1} := (1-\tau_k)\gamma_k$, i.e., $c_k = 1$. We can also fix $\gamma_k = \gamma_0 > 0$ for all the iterations $k \geq 0$, i.e., $c_k = 0$. It is important to note that Step 5 and Step 6 are mixed. Depending on the use of either (2P1D) or (1P2D), the corresponding parameter β_k or γ_k is updated before Step 5. If we choose $c_k < 0$, then $\{\gamma_k\}$ is increasing. Since the rate of $\beta_k\gamma_k$ is fixed at $\mathcal{O}(1/k^2)$ due to (44), if we decrease the rate of $\{\gamma_k\}$ (i.e., increase γ_k), then $\{\beta_k\}$ converges faster than the $\mathcal{O}(1/k^2)$ rate. We will discuss the stopping condition at Step 4 later. We note that we can also alternate between (2P1D) and (1P2D) in Algorithm 1. However, it is not clear whether this strategy would yield any numerical advantage.

4.6 Convergence analysis

Under Assumption 1, the dual solution set $\mathscr{Y}^\star$ is nonempty. Recall that $D_{\mathscr{Y}^\star} := \min_{\mathbf{y}^\star \in \mathscr{Y}^\star} \|\mathbf{y}^\star\|_2 < +\infty$ is the norm of a minimum norm dual solution. The following theorem shows the convergence of Algorithm 1.

Theorem 4.1 *Let $\left\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ be the sequence generated by Algorithm 1 after $k \geq 1$ iterations. Then, if $g_\gamma \equiv \tilde{g}_\gamma$, i.e., using augmented Lagrangian smoother $\tilde{g}_\gamma$, then:*

a) *If $c_k := 0$ for all $k \geq 0$, $\gamma_0 := \bar{L}^{\tilde{g}} = 1$, then:*

$$\left\{\begin{array}{ll} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{8D_{\mathscr{Y}^\star}}{(k+1)^2}, \\ -\frac{1}{2}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2 - D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^k) - f^\star \leq 0, \end{array}\right. \tag{46}$$

for all $k \geq 0$. Moreover, the spectral norm of $\mathbf{A}$ does not affect the bounds in (46)*.*

As a consequence, the worst-case analytical complexity of Algorithm 1 to achieve an ε-primal solution $\bar{\mathbf{x}}^k$ for (1) *in the sense of Definition 2.1 is $\mathscr{O}\left(\varepsilon^{-1/2}\right)$.*

Alternatively, if $g_\gamma \equiv \hat{g}_\gamma$, i.e., using Bregman distance smoother $\tilde{g}_\gamma$, then:

b) *If Algorithm 1 uses* (2P1D)*, $\gamma_0 := \sqrt{\bar{L}^g}$ and $c_k := 1$ for all $k \geq 0$, then:*

$$\left\{\begin{array}{ll} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{\sqrt{\bar{L}^g}\left(2D_{\mathscr{Y}^\star} + \sqrt{2D^{\mathbb{I}}_{\mathscr{X}}}\right)}{k+1}, \\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^k) - f^\star \leq \frac{\sqrt{\bar{L}^g}}{k+1} D^{\mathbb{I}}_{\mathscr{X}}. \end{array}\right. \tag{47}$$

c) *If Algorithm 1 uses* (1P2D)*, $\gamma_0 := \frac{2\sqrt{2\bar{L}_g}}{K+1}$ and $c_k := 0$ for all $k = 0, \ldots, K$, then:*

$$\left\{\begin{array}{ll} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|_2 \leq \frac{2\sqrt{2\bar{L}_g}(D_{\mathscr{Y}^\star} + \sqrt{D^{\mathbb{I}}_{\mathscr{X}}})}{(K+1)}, \\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^K) - f^\star \leq \frac{2\sqrt{2\bar{L}_g}}{(K+1)} D^{\mathbb{I}}_{\mathscr{X}}. \end{array}\right. \tag{48}$$

As a consequence, the worst-case analytical complexity of Algorithm 1 to achieve an ε-primal solution $\bar{\mathbf{x}}^k$ for (1) *in the sense of Definition 2.1 is $\mathscr{O}\left(\varepsilon^{-1}\right)$.*

We note that the choice of γ_0 in Theorem 4.1 trades-off the primal objective residual and the primal feasibility gap. Indeed, smaller γ_0 leads to smaller $|f(\bar{\mathbf{x}}^k) - f^\star|$.

We chose the (1P2D) scheme above due to its close relationship to some well-known primal dual methods we describe below. Unfortunately, the (1P2D) scheme has the drawback of fixing the total number of iterations *a priori*, which the (2P1D) scheme can avoid at the expense of more proximal operator calculations.

5 Instances of Algorithm 1

This section specifies Algorithm 1 under different assumptions to obtain specific instances of this algorithm for solving (1).

5.1 Strong convexity assumption

If the objective function f of (1) is strongly convex with a convexity parameter $\sigma_f > 0$. Then it is well-known that (see, e.g., [50]) the dual function $g(\cdot)$ defined by (10) is smooth and Lipschitz gradient with a Lipschitz constant $L_f^g := \frac{\|\mathbf{A}\|_2^2}{\sigma_f}$. In this case, we modify accordingly both schemes (2P1D) and (1P2D) as follows:

$$(\text{2P1D}_\sigma)\left\{\begin{array}{ll}\hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star(\bar{\mathbf{y}}^k),\\ \bar{\mathbf{x}}^{k+1} & := \text{prox}_{\mathbb{I}_f}(\hat{\mathbf{x}}^k, \beta_k^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}); \beta_k),\\ \bar{\mathbf{y}}^{k+1} & := (1-\tau_k)\bar{\mathbf{y}}^k + \frac{\tau_k}{\beta_k}\big(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\big).\end{array}\right. \qquad (\text{1P2D}_\sigma)\left\{\begin{array}{ll}\hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k \mathbf{y}^\star_{\beta_k}(\bar{\mathbf{x}}^k),\\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star(\hat{\mathbf{y}}^k),\\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + \frac{1}{L_f^g}\big(\mathbf{A}\mathbf{x}^\star(\hat{\mathbf{y}}^k) - \mathbf{b}\big).\end{array}\right.$$

While the scheme (1P2D_σ) remains similarly to (1P2D), the parameter β_k in (2P1D_σ) has not updated yet as in (2P1D).

The starting point $\bar{\mathbf{w}}^0 := (\mathbf{x}^\star(0^m), \bar{\mathbf{y}}^0) \in \mathscr{W}$ for Algorithm 1 with respect to this variant can be computed as $\bar{\mathbf{y}}^0 := (L_f^g)^{-1}(\mathbf{A}\mathbf{x}^\star(0^m) - \mathbf{b})$ and $\mathbf{x}^\star(\mathbf{y})$ is the unique solution of the minimization in (10). The parameters β_k and τ_k are updated as follows:

$$\beta_{k+1} := (1-\tau_k)\beta_k, \quad \tau_{k+1} := (\tau_k/2)[(\tau_k^2+4)^{1/2} - \tau_k], \quad k \geq 0, \tag{49}$$

where $\beta_0 := L_f^g$ and $\tau_0 := (\sqrt{5}-1)/2$. The following corollary shows the convergence of both schemes, whose proof is in the appendix.

Corollary 5.1 *Assume that f of* (1) *is σ_f-strongly convex. Let $\big\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\big\}_{k\geq 0}$ be a sequence generated by either* (2P1D_σ) *or* (1P2D_σ) *using the update rule* (49)*. Then*

$$\left\{\begin{array}{lll} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & \frac{4\|\mathbf{A}\|_2^2}{(k+2)^2\sigma_f} D_{\mathscr{Y}^\star},\\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^k) - f^\star \leq & 0,\\ & \|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2 \leq & \frac{4\|\mathbf{A}\|_2}{(k+2)\sigma_f} D_{\mathscr{Y}^\star},\end{array}\right. \tag{50}$$

where $D_{\mathscr{Y}^\star}$ is defined in Theorem 4.1 and $\mathbf{x}^\star \in \mathscr{X}^\star$.

As a consequence, the worst-case analytical complexity for finding an ε-primal solution $\bar{\mathbf{x}}^k$ of (1) *in the sense of Definition 2.1 is $\mathcal{O}(1/\sqrt{\varepsilon})$.*

Remark 5.1 *The bounds in* (50) *do not depend on the prox-diameter $D_{\mathscr{X}}^{\mathbb{I}}$ of the feasible set $\mathscr{X}$. Hence, the boundedness of $\mathscr{X}$ is no longer required.*

Remark 5.2 *Convergence of the objective indeed depends on the absolute value of the primal residual, i.e., $|f(\bar{\mathbf{x}}^k) - f^\star| \leq \frac{4\|\mathbf{A}\|_2^2}{(k+2)^2\sigma_f} D_{\mathscr{Y}^\star}^2$.*

5.2 Lipschitz gradient assumption

The aim of this subsection is to develop a variant of Algorithm 1 using (1P2D) without fixed the accuracy as stated in Theorem 4.1(c). However, this variant is only limited to problems of the form (1) that satisfy the following technical assumption:

Assumption A. 2 *The following conditions hold:*

(a) The objective function f and the feasible set $\mathscr{X}$ of (1) *are separable as in* (3)*.*

(b) The last term f_p is L_{f_p}-Lipschitz gradient and the smallest eigenvalue $\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p)$ of matrix $\mathbf{A}_p$ is positive.

(c) The Bregman distance $d(\mathbf{Sx},\mathbf{Sx}_c)$ *is chosen as* $d(\mathbf{Sx},\mathbf{Sx}_c) := \sum_{i=1}^p d_i(\mathbf{S}_i\mathbf{x}_i,\mathbf{S}_i\mathbf{x}_i^c)$*, where* $\mathbf{S}_p \equiv \mathbb{I}$ *and* $d_p(\cdot,\mathbf{x}_p^c)$ *is smooth and* $\nabla d_p(\cdot,\mathbf{x}_p^c)$ *is* 1*-Lipschitz continuous.*

(d) The last term g_γ^p *of the smoothed dual function* g_γ *defined by* (17) *satisfies*

$$g_\gamma^p(\mathbf{y}) = \min_{\mathbf{x}_p\in\mathbb{R}^{n_p}} \left\{ f_p(\mathbf{x}_p) + \mathbf{y}^T\mathbf{A}_p\mathbf{x}_p + (\gamma/2)d_p(\mathbf{x}_p,\mathbf{x}_p^c) \right\}. \tag{51}$$

That is the primal constraint on the last component is not active.

Under Assumption A.2, we can write the function g_γ defined by (17) as $g_\gamma(\mathbf{y}) := \sum_{i=1}^p g_\gamma^i(\mathbf{y}) - \mathbf{b}^T\mathbf{y}$, where

$$g_\gamma^i(\mathbf{y}) := \min_{\mathbf{x}_i\in\mathscr{X}_i} \left\{ f_i(\mathbf{x}_i) + \mathbf{y}^T\mathbf{A}_i\mathbf{x}_i + (\gamma/2)d_i(\mathbf{S}_i\mathbf{x}_i,\mathbf{S}_i\mathbf{x}_i^c) \right\}, \quad i = 1,\dots,p.$$

A simple example for d_p is $d_p(\mathbf{x}_p) := (1/2)\|\mathbf{x}_p - \mathbf{x}_p^c\|_2^2$. The last condition in Assumption A.2 shows that the solution $\mathbf{x}_{p,\gamma}^\star(\mathbf{y})$ of the minimization problem in g_γ^p must be attained in $\operatorname{relint}(\mathbf{X}_p)$. This condition is not too restrictive, since we only require it for the last component g_γ^p. It is automatically fulfilled if f_p is strongly convex and $\mathbf{x}_p^c \in \operatorname{relint}(\mathbf{X}_p)$. Now, we show that the function g_γ^p is strongly concave in the following lemma, whose proof can be found in the appendix.

Lemma 5.1 *Under Assumption A.2, the function* g_γ^p *defined by* (51) *is strongly concave with the parameter* $\sigma_{g_\gamma^p} := (L_{f_p}+\gamma)^{-1}\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p) > 0$*. Consequently, the function* g_γ *defined by* (17) *is also strongly convex with the same parameter* $\sigma_{g_\gamma^p}$*.*

Using the result of Lemma 5.1, we can update γ_k and β_k in the scheme (1P2D) as

$$\gamma_{k+1} := \big(1-\tau_k/(1+\tau_k)\big)\gamma_k, \;\; \beta_{k+1} := (1-\tau_k)\beta_k \;\text{ and }\; \tau_k := (k+1)^{-1} \;\; \forall k \geq 0, \tag{52}$$

where $\beta_0 = \gamma_0 := \sqrt{\bar{L}^g}$. In this case, we have $\gamma_{k+1}\beta_{k+1} \geq \bar{L}^g\tau_k^2$ for $k \geq 0$. The following corollary shows the convergence of this variant.

Corollary 5.2 *Under Assumption A.2, let* $\left\{(\bar{\mathbf{x}}^k,\bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ *be a sequence generated by* (1P2D) *using the update rule* (52)*. Then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \;\; \leq \frac{2\sqrt{2\bar{L}^g}\left(D_{\mathscr{Y}^\star} + \sqrt{D_{\mathscr{X}}^{\mathbf{S}}}\right)}{k+1}, \\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^k) - f^\star \;\; \leq \frac{2\sqrt{2\bar{L}^g}}{k+1} D_{\mathscr{X}}^{\mathbf{S}}, \end{cases} \tag{53}$$

where $D_{\mathscr{Y}^\star}$ *and* $\bar{L}^g$ *are defined in Theorem 4.1.*

Remark 5.3 *Corollary 5.2 shows that, for certain subclass of problems* (1) *satisfying Assumption A.2, it allows us to simultaneously update both parameters* γ_k *and* β_k *instead of fixing a priori* γ_0 *as in Theorem 4.1(c).*

5.3 Inexact solution of the augmented Lagrangian smoother

In the augmented Lagrangian smoothing method, solving the minimization problem (19) exactly can be impracticable. However, we can often solve this subproblem up to a given accuracy $\delta > 0$, i.e.,

$$\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}) := \delta\text{-}\arg\min_{\mathbf{x}\in\mathscr{X}} \left\{ \mathscr{L}_\gamma(\mathbf{x},\mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^T(\mathbf{Ax}-\mathbf{b}) + (\gamma/2)\|\mathbf{Ax}-\mathbf{b}\|_2^2 \right\}, \tag{54}$$

in the following sense:

$$\mathscr{L}_\gamma(\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}),\mathbf{y}) - \mathscr{L}_\gamma(\tilde{\mathbf{x}}_\gamma^\star(\mathbf{y}),\mathbf{y}) \leq \gamma\delta^2/2, \tag{55}$$

where $\tilde{\mathbf{x}}_\gamma^\star(\mathbf{y})$ is an exact solution of (19).

The condition $\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}) \in \mathscr{X}$ is reasonable in practice since the feasible set $\mathscr{X}$ can be assumed to be "simple" so that the computation of the projection onto $\mathscr{X}$ can be carried out exactly. In addition, there exist several convex optimization algorithms (e.g., Nesterov's accelerated algorithms [48]) for computing $\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y})$ that satisfy (55).

By the definition of $\mathscr{L}_\gamma$, we can easily show that

$$\mathscr{L}_\gamma(\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}),\mathbf{y}) - \mathscr{L}_\gamma(\tilde{\mathbf{x}}_\gamma^\star(\mathbf{y}),\mathbf{y}) \geq (\gamma/2)\|\mathbf{A}(\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}) - \tilde{\mathbf{x}}_\gamma^\star(\mathbf{y}))\|_2^2,$$

which leads to $\|\mathbf{A}(\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}) - \tilde{\mathbf{x}}_\gamma^\star(\mathbf{y}))\|_2 \leq \delta$. Now, if we define $\nabla\tilde{g}_\gamma^\delta(\mathbf{y}) := \mathbf{A}\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}) - \mathbf{b}$ an approximation for the gradient $\nabla\tilde{g}_\gamma(\mathbf{y})$, then (55) and the last inequality implies

$$\|\nabla\tilde{g}_\gamma^\delta(\mathbf{y}) - \nabla\tilde{g}_\gamma(\mathbf{y})\|_2 \leq \delta. \tag{56}$$

In addition, we also denote by $\tilde{g}_\gamma^\delta(\mathbf{y}) := \tilde{\mathscr{L}}_\gamma(\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y}),\mathbf{y})$ as an approximation to $\tilde{g}_\gamma(\mathbf{y})$.

Instead of using the true solution $\mathbf{x}_\gamma^\star(\mathbf{y})$ in the schemes (2P1D) and (1P2D), we use the approximate solutions $\tilde{\mathbf{x}}_\gamma^\delta(\mathbf{y})$ to obtain the following inexact iterative schemes:

$$\boxed{\begin{array}{c}\text{(i2P1D)}\\ \left\{\begin{array}{ll}\hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\tilde{\mathbf{x}}_{\gamma_k}^{\delta_k}(\bar{\mathbf{y}}^k),\\ \hat{\mathbf{y}}^k & := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}),\\ \bar{\mathbf{x}}^{k+1} & := \widetilde{\text{prox}}_{\mathbf{A}f}^{\delta_k}(\hat{\mathbf{x}}^k,\hat{\mathbf{y}}^k;\beta_{k+1}),\\ \bar{\mathbf{y}}^{k+1} & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\hat{\mathbf{y}}^k.\end{array}\right.\end{array}} \qquad \boxed{\begin{array}{c}\text{(i1P2D)}\\ \left\{\begin{array}{ll}\bar{\mathbf{y}}_k^\star & := \beta_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}),\\ \hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\bar{\mathbf{y}}_k^\star,\\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\tilde{\mathbf{x}}_{\gamma_k}^{\delta_k}(\hat{\mathbf{y}}^k),\\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + \gamma_k\big(\mathbf{A}\tilde{\mathbf{x}}_{\gamma_k}^{\delta_k}(\hat{\mathbf{y}}^k) - \mathbf{b}\big).\end{array}\right.\end{array}} \tag{57}$$

Here, the inexact proximal operator $\widetilde{\text{prox}}_{\mathbf{A}f}^\delta$ is defined as:

$$\widetilde{\text{prox}}_{\mathbf{A}f}^\delta(\bar{\mathbf{x}},\hat{\mathbf{y}};\beta) := \delta\text{-}\arg\min_{\mathbf{x}\in\mathscr{X}}\left\{\mathscr{H}_\beta(\mathbf{x};\hat{\mathbf{y}},\bar{\mathbf{x}}) := f(\mathbf{x}) + \hat{\mathbf{y}}^T\mathbf{A}(\mathbf{x}-\bar{\mathbf{x}}) + \frac{\bar{L}^g}{2\beta}\|\mathbf{A}(\mathbf{x}-\bar{\mathbf{x}})\|_2^2\right\}, \tag{58}$$

where $\mathbf{A}$ and $\delta \geq 0$ are given and the inexactness is also defined as in (55).

The starting point $\bar{\mathbf{w}}^0 := (\bar{\mathbf{x}}^0,\bar{\mathbf{y}}^0) \in \mathscr{W}$ can be computed from one of the following formulations:

$$\left\{\begin{array}{ll}\bar{\mathbf{x}}^0 & := \tilde{\mathbf{x}}_{\gamma_0}^{\delta_0}(0^m),\\ \bar{\mathbf{y}}^0 & := \beta_0^{-1}\left(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\right),\end{array}\right. \quad \text{or} \quad \left\{\begin{array}{ll}\bar{\mathbf{y}}^0 & := \beta_0^{-1}\left(\mathbf{A}\mathbf{x}_c - \mathbf{b}\right),\\ \bar{\mathbf{x}}^0 & := \widetilde{\text{prox}}_{\mathbf{A}f}^{\delta_0}(\mathbf{x}_c,\bar{\mathbf{y}}^0;\beta_0).\end{array}\right. \tag{59}$$

The following theorem shows the convergence of the inexact variant of Algorithm 1 using scheme (57), called (i1P2D), whose proof can be found in the appendix. Analogously, we can also prove the same result as in Theorem 5.1 for the (i2P1D) scheme but we omit the laborious details.

Theorem 5.1 *Let $\left\{(\bar{\mathbf{x}}^k,\bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ be the sequence generated by Algorithm 1 using* (i1P2D) *in* (57) *and the first initial point $(\bar{\mathbf{x}}^0,\bar{\mathbf{y}}^0)$ in* (59)*. Then, if $\gamma_0 = \bar{L}^{\tilde{g}} = 1$, $c_k := 0$ and $q_k\delta_k \leq q_{k-1}\delta_{k-1}$ for all $k \geq 0$ and $q_k := (1-\tau_k)\tau_k\|\bar{\mathbf{y}}^k - \bar{\mathbf{y}}_k^\star\|_2 + (D_{\mathscr{X}}^{\mathbf{A}}+1)/2$ then:*

$$\left\{\begin{array}{lll} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 & \leq \frac{4}{(k+1)^2}\left(2D_{\mathscr{Y}^\star} + \sqrt{\frac{14q_0\delta_0}{(k+1)^2}}\right),\\ -(1/2)\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2 - D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^k) - f^\star & \leq 7q_0\delta_0.\end{array}\right. \tag{60}$$

As a consequence, if $\delta_0 = \mathscr{O}\left(\frac{q_0}{k^2}\right)$, then the worst-case analytical complexity of Algorithm 1 to achieve an ε-primal solution $\bar{\mathbf{x}}^k$ of (1) *in the sense of Definition 2.1 is $\mathscr{O}\left(\varepsilon^{-1/2}\right)$.*

Theorem 5.1 shows that the primal feasibility gap $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$ converges to 0^+ at the rate $\mathscr{O}(1/k^2)$, while the objective residual $\left|f(\bar{\mathbf{x}}^k) - f^\star\right|$ depends on the numerical accuracy δ_0 of (54) at the initial iteration $k = 0$. If δ_0 is not sufficiently small, we only obtain a sub-optimal solution of (1). Practically, we can solve (54) at $k = 0$ with relatively high accuracy and use a warm-start strategy to significantly reduce the computational burden of the subsequent iterations.

6 Explicit connections to existing methods

To better differentiate our contributions, it is important to make explicit comparisons of Algorithm 1 with the dual fast gradient methods, alternating direction methods of multipliers (ADMM) and proximal-based decomposition methods here.

6.1 Connections to the fast gradient methods

Dual fast gradient methods were studied in, e.g., [5, 44, 45, 59]. The main idea is to use either the strong convexity of the objective [5, 59] or smoothing technique via prox-functions [44] or augmented Lagrangian function [45], which leads to the Lipschitz continuity of the gradient of the dual function. Then, Nesterov's fast gradient method [48] is applied to solve the smoothed dual problem.

In this paper, we also smooth the dual function by using either augmented Lagrangian function or Bregman distances to obtain a smoothed dual function with Lipschitz continuous gradient. In order to obtain both primal objective residual and primal feasibility gap simultaneously, we exploit the concepts of excessive gap technique introduced by Nesterov [49] and Auslander's gap function [1] to build a primal-dual sequence $\left\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ that converges to the primal-dual optimal solution $(\mathbf{x}^\star, \mathbf{y}^\star)$ of (1). In [44, 59] the authors only proved the convergence results in terms of the dual objective values g, which is different from Theorem 4.1, where we both have the convergence rate guarantee both on the primal objective residual and the primal feasibility gap. In [45] the authors characterized the convergence rate of an inexact augmented Lagrangian method both in the primal objective values and the primal feasibility gaps. However, the approach is directly based on Nesterov's accelerated scheme for the dual problem and the convergence results are presented in an ergodic sense. In [5] the authors considered a special case of (1), where the objective function is strongly convex as in Corollary 5.1. They also characterized the feasibility gap. However, the convergence rate of this quantity drops to $\mathscr{O}(1/k)$ instead of the better $\mathscr{O}(1/k^2)$ rate established by our Corollary 5.1.

We close this discussion by showing that our results in Corollary 5.1 can be applied to non-strongly convex problems of the form (1). We process this procedure as follows. Assume that f of (1) is not strongly convex, we consider the function $f_\sigma(\mathbf{x}) := f(\mathbf{x}) + (\sigma_f/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$, where $\sigma_f > 0$ and $\mathbf{x}_c \in \mathscr{X}$. Then, the function f_σ is strongly convex with the parameter $\sigma_f > 0$. Next, we apply either (2P1D$_\sigma$) or (1P2D$_\sigma$) to solve (1) with f is substituted by f_σ. In this case, Corollary 5.1 is still valid. Moreover, we have $f_\sigma(\bar{\mathbf{x}}^k) = f(\bar{\mathbf{x}}^k) + (\sigma_f/2)\|\bar{\mathbf{x}}^k - \mathbf{x}_c\|_2^2$ and $f_\sigma^\star = f^\star + (\sigma_f/2)\|\mathbf{x}^\star - \mathbf{x}_c\|_2^2$, which imply

$$|f(\bar{\mathbf{x}}^k) - f^\star| \leq |f_\sigma(\bar{\mathbf{x}}^k) - f_\sigma^\star| + 2\sigma_f D_{\mathscr{X}}^{\mathbb{I}},$$

where $D_{\mathscr{X}}^{\mathbb{I}} := \max_{\mathbf{x}\in\mathscr{X}}(1/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$. Combining this estimate and Corollary 5.1 we obtain $|f(\bar{\mathbf{x}}^k) - f^\star| \leq \frac{4\|\mathbf{A}\|_2^2}{\sigma_f(k+2)^2} D_{\mathscr{Y}^\star}^2 + 2\sigma_f D_{\mathscr{X}}^{\mathbb{I}}$. Hence, if we choose $\sigma_f := \frac{\sqrt{2}\|\mathbf{A}\|_2 D_{\mathscr{Y}^\star}}{(k+2)\sqrt{D_{\mathscr{X}}^{\mathbb{I}}}}$ then we obtain the worst-case analytical complexity of this algorithm as

$$|f(\bar{\mathbf{x}}^k) - f^\star| \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2 D_{\mathscr{Y}^\star}(D_{\mathscr{X}}^{\mathbb{I}})^{1/2}}{(k+2)} \;\text{ and }\; \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2 (D_{\mathscr{X}}^{\mathbb{I}})^{1/2}}{(k+2)}.$$

Comparing this complexity and Theorem 4.1, we conclude that depending on the values of $D_{\mathscr{Y}}^\star$ and $D_{\mathscr{X}}^{\mathbb{I}}$ we can use choose an appropriate variant of Algorithm 1 for solving the given problem. However, note that we do

not generally have access to $D_{\mathscr{Y}^\star}$, hence we can instead use the standard (1P2D) or (2D1P) schemes which do not require the knowledge of the smoothing parameter.

6.2 Connections to ADMMs

Several algorithms based on method of multipliers such as alternating minimization algorithm (AMA) [68], alternating direction method of multipliers (ADMM) [11] and alternating linearization methods (ALM) [30] have been developed in the literature. Such methods initially aim at solving instances of (1) when f and $\mathscr{X}$ are separable with $p = 2$ as defined as (3). In this case, the primal step (17) or (34) is computed by solving two subproblems with respect to $\mathbf{x}_1$ and $\mathbf{x}_2$ alternatively.

Let $f(\mathbf{x}) := f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$ and $\mathscr{X} := \mathscr{X}_1 \times \mathscr{X}_2$. To compare with our algorithm, we rewrite the standard ADMM algorithm [68] as follows:

$$\left\{\begin{array}{ll} \mathbf{x}_1^{k+1} & := \underset{\mathbf{x}_1 \in \mathscr{X}_1}{\operatorname{argmin}} \left\{ f_1(\mathbf{x}_1) + (\gamma_k/2) \left\| \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b} + \gamma_k^{-1}\mathbf{y}^k \right\|_2^2 \right\}, \\ \mathbf{x}_2^{k+1} & := \underset{\mathbf{x}_2 \in \mathscr{X}_2}{\operatorname{argmin}} \left\{ f_2(\mathbf{x}_2) + (\gamma_k/2) \left\| \mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} + \gamma_k^{-1}\mathbf{y}^k \right\|_2^2 \right\}, \\ \mathbf{y}^{k+1} & := \mathbf{y}^k + \gamma_k(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{b}), \end{array}\right. \tag{61}$$

where $\gamma_k > 0$ is a given penalty parameter.

Let us modify the (1P2D) scheme by using the primal step as in (61) to obtain:

$$\left\{\begin{array}{ll} \hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\beta_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}), \\ \mathbf{x}_1^{k+1} & := \underset{\mathbf{x}_1 \in \mathscr{X}_1}{\operatorname{argmin}} \left\{ f_1(\mathbf{x}_1) + (\gamma_k/2) \left\| \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b} + \gamma_k^{-1}\mathbf{y}^k \right\|_2^2 \right\}, \\ \mathbf{x}_2^{k+1} & := \underset{\mathbf{x}_2 \in \mathscr{X}_2}{\operatorname{argmin}} \left\{ f_2(\mathbf{x}_2) + (\gamma_k/2) \left\| \mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} + \gamma_k^{-1}\mathbf{y}^k \right\|_2^2 \right\}, \\ \bar{\mathbf{x}}_{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\mathbf{x}^{k+1}, \\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + (\gamma_k/2)(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{b}). \end{array}\right. \tag{62}$$

It is trivial that if $\tau_k = 0$ then (62) coincides with the standard ADMM scheme (61). We also fix the parameter γ_k at an appropriate value γ_0 sufficiently small, while update only the parameter β_k as $\beta_{k+1} := (1-\tau_k)\beta_k$. The following corollary shows the convergence of the new ADMM variant (62), whose proof can be found in the appendix.

Corollary 6.1 *Let $\left\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ be a sequence generated by Algorithm 1 using the new* ADMM *variant* (62). *If $\gamma_k \equiv \gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|_2}{K+3}$ and $\beta_{k+1} := (1-\tau_k)\beta_k$ for all $k = 0, 1, \ldots, K$, then:*

$$\left\{\begin{array}{lll} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|_2 & \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2(D_{\mathscr{Y}^\star} + \bar{D}_{\mathscr{X}}^{\mathbf{A}})}{(K+3)}, \\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|_2 \leq & f(\bar{\mathbf{x}}^K) - f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2}{(K+3)}(\bar{D}_{\mathscr{X}}^{\mathbf{A}})^2, \end{array}\right. \tag{63}$$

where $\bar{D}_{\mathscr{X}}^{\mathbf{A}} := 2\max\{\|\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}})\|_2 \mid \mathbf{x}, \hat{\mathbf{x}} \in \mathscr{X}\}$.

As a consequence, the worst-case analytical complexity of Algorithm 1 to achieve an ε-primal solution $\bar{\mathbf{x}}^k$ of (1) *in the sense of Definition 2.1 is $\mathscr{O}\left(\varepsilon^{-1}\right)$.*

We note that solving two primal subproblems in (62) is still a challenge in practice if $\mathbf{A}_1$ and $\mathbf{A}_2$ are dense, large-scale linear operators. If f_1 and f_2 have tractable proximal operators $\operatorname{prox}_{\lambda f}$ as defined in (4), then instead of solving two minimization problems in (61), we can linearize the quadratic term to obtain a preconditioned ADMM (PADMM) as considered in [15]. Indeed, PADMM can be viewed as a variant of Chambolle-Pock's primal-dual algorithm [15]. This algorithm can also be seen as a variant of the primal-dual hybrid gradient

algorithm (PDHG) considered in [27, 31]. In our setting (62), we linearize the primal step to obtain the following new PADMM variant of the ADMM scheme (62):

$$\left\{\begin{array}{ll} \mathbf{x}_1^{k+1} & := \arg\min\limits_{\mathbf{x}_1\in\mathscr{X}_1}\left\{f_1(\mathbf{x}_1)+(\gamma_k/(2\alpha_{1k}))\|\mathbf{x}_1-(\mathbf{g}_1^k+\gamma_k^{-1}(\mathbf{A}_1^T\mathbf{y}^k)\|_2^2\right\},\\ \mathbf{x}_2^{k+1} & := \arg\min\limits_{\mathbf{x}_2\in\mathscr{X}_2}\left\{f_2(\mathbf{x}_2)+(\gamma_k/(2\alpha_{2k}))\|\mathbf{x}_2-(\mathbf{g}_2^k+\gamma_k^{-1}(\mathbf{A}_2^T\mathbf{y}^k))\|_2^2\right\},\end{array}\right. \tag{64}$$

where $\mathbf{g}_1^k := \mathbf{x}_1^k - \alpha_{1k}\mathbf{A}_1^T(\mathbf{A}_1\mathbf{x}_1^k+\mathbf{A}_2\mathbf{x}_2^k-\mathbf{b})$, $\mathbf{g}_2^k := \mathbf{x}_2^k - \alpha_{2k}\mathbf{A}_2^T(\mathbf{A}_1\mathbf{x}_1^{k+1}+\mathbf{A}_2\mathbf{x}_2^k-\mathbf{b})$, and step size α_{1k} and α_{2k} are chosen from gradient methods [71].

Similarly to Corollary 6.1, the following corollary shows the convergence of the new PADMM scheme (62) (a variant of (64)), whose proof can also be found in the appendix.

Corollary 6.2 *Let $\left\{(\bar{\mathbf{x}}^k,\bar{\mathbf{y}}^k)\right\}_{k\geq 0}$ be a sequence generated by Algorithm 1 using the* PADMM *scheme* (62)-(64). *If $\gamma_k\equiv\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|_2}{K+3}$ and $\beta_{k+1} := (1-\tau_k)\beta_k$ for all $k=0,1,\ldots,K$, then:*

$$\left\{\begin{array}{lll} & \|\mathbf{A}\bar{\mathbf{x}}^K-\mathbf{b}\|_2 & \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2(D_{\mathscr{Y}^\star}+D^{\mathbb{I}}_{\mathscr{X}})}{(K+3)},\\ -D_{\mathscr{Y}^\star}\|\mathbf{A}\bar{\mathbf{x}}^K-\mathbf{b}\|_2\leq & f(\bar{\mathbf{x}}^K)-f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|_2}{(K+3)}(D^{\mathbb{I}}_{\mathscr{X}})^2,\end{array}\right. \tag{65}$$

where $D^{\mathbb{I}}_{\mathscr{X}} := 4\max\{\|\mathbf{x}-\hat{\mathbf{x}}\|_2 \mid \mathbf{x},\ \hat{\mathbf{x}}\in\mathscr{X}\}$.

As a consequence, the worst-case analytical complexity of Algorithm 1 to achieve an ε-primal solution $\bar{\mathbf{x}}^k$ of (1) *in the sense of Definition 2.1 is $\mathscr{O}\left(\varepsilon^{-1}\right)$.*

Although Corollaries 6.1 and 6.2 prove the optimal convergence rate of the new ADMM and PADMM variants, respectively, they still require to specify the total number of iterations K *a priori* to choose γ_0, which is a drawback of our current analysis.

In [37, 38], the authors proved the convergence of the standard ADMM algorithm at the rate of $\mathscr{O}(1/k)$ but in the sense of Auslender's gap function and requires the boundedness of both the primal and dual feasible sets. More precisely, they showed that [38, Theorem 4.1.] $G(\tilde{\mathbf{w}}^k)\leq\frac{D^2_{\mathscr{W}}}{2(k+1)}$, where $D_{\mathscr{W}} := \sup_{\mathbf{w}\in\mathscr{W}}\|\mathbf{w}-\mathbf{w}^0\|_2$ and $\tilde{\mathbf{w}}^k := (k+1)^{-1}\sum_{j=0}^k\mathbf{w}^j$ and $\mathbf{w}^k := (\mathbf{x}^k,\mathbf{y}^k)$ generated by (61). In [57] the authors considered other variant of ADMM, which requires the Lipschitz gradient assumption of f_1 or f_2 and still obtained the $\mathscr{O}(1/k)$ convergence rate both on the objective residual $f(\mathbf{x}^k)-f^\star$ and the feasibility gap $\|\mathbf{A}\mathbf{x}^k-\mathbf{b}\|_2$. Other variants of ADMM can be found, e.g., in [23, 56, 70] and the references quoted therein, which were applied to stochastic cases or using different set of assumptions. However, the main steps in these variants principally remain the same as (61).

6.3 Connections to proximal-based decomposition method

If we set $\mathbf{x}_c^k\equiv\bar{\mathbf{x}}^{k-1}$ for $k\geq 1$ in our (1P2D) scheme, then the resulting scheme closely relates to the proximal-based decomposition method (PBDM) studied in [20, 64]. Indeed, the main steps of PBDM can be expressed as follows:

$$\left\{\begin{array}{ll} \hat{\mathbf{y}}^k & := \bar{\mathbf{y}}^k+\gamma_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}),\\ \mathbf{x}_1^{k+1} & := \arg\min\limits_{\mathbf{x}_1\in\mathscr{X}_1}\left\{f_1(\mathbf{x}_1)+(\hat{\mathbf{y}}^k)^T\mathbf{A}_1\mathbf{x}_1+(\gamma_k/2)\|\mathbf{x}_1-\mathbf{x}_1^k\|_2^2\right\},\\ \mathbf{x}_2^{k+1} & := \arg\min\limits_{\mathbf{x}_2\in\mathscr{X}_2}\left\{f_2(\mathbf{x}_2)+(\hat{\mathbf{y}}^k)^T\mathbf{A}_2\mathbf{x}_2+(\gamma_k/2)\|\mathbf{x}_2-\mathbf{x}_2^k\|_2^2\right\},\\ \bar{\mathbf{y}}^{k+1} & := \bar{\mathbf{y}}^k+\gamma_k^{-1}(\mathbf{A}_1\mathbf{x}_1^{k+1}+\mathbf{A}_2\mathbf{x}_2^{k+1}-\mathbf{b}).\end{array}\right. \tag{66}$$

Clearly, this method looks very similar to (1P2D), where it has two dual steps and one primal step. Here, (66) uses only one parameter γ_k, $d_b(\mathbf{x},\hat{\mathbf{x}}) := \frac{1}{2}\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ the Euclidian distance and $\mathbf{S}\equiv\mathbb{I}$. In [64] the authors prove the

convergence of the scheme (66) in a joint criterion $f(\tilde{\mathbf{x}}^k) - f^\star + r\|\mathbf{A}\tilde{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{1}{k+1}\left[c_1 + c_2 \max_{\|\mathbf{y}\|\leq r}\|\mathbf{y}-\mathbf{y}^0\|_2^2\right]$, where $\tilde{\mathbf{x}}^k := \frac{1}{k}\sum_{j=0}^{k-1}\mathbf{x}^{j+1}$ and c_1, c_2 and r are given constants. This result is very similar to the ones in [37, 38] for ADMM, which combines the primal objective residual and the primal feasibility gap. However, since (1) is constrained, $f(\mathbf{x}^k) - f^\star$ may take an arbitrarily negative value. Hence, the joint criterion does not imply the approximation of the primal objective residual and the primal feasibility gap separately. Moreover, as indicated in [31], convergence guarantee in a joint criterion is not sufficient to ensure that primal-dual methods work well in practice. It is important to control algorithmic parameters to trade-off between the objective residual and the feasibility of the problem. In our case, we prove a separated criterion on the objective residual and the primal feasibility, which allows one to control the parameters in order to trade-off these quantities. At the same time, our methods still exploit p-decomposability with parallel updates in the primal steps (17) and (34).

7 Implementation enhancements

We discuss in this section how to enhance the practical performance of Algorithm 1. We observe that at least three steps in Algorithm 1 can be modified to enhance its practical performance: the choice of $\mathbf{x}_c^k$, the update rule for parameters as well as the parallel and distributed implementation choices.

7.1 The choice of proximal-point $\mathbf{x}_c^k$ and Bregman distances

In (2P1D) and (1P2D), we can adaptively choose the center point $\mathbf{x}_c^k$ of the Bregman distance at each iteration. We propose two options:

- *Proximal-point:* We can choose $\mathbf{x}_c^k := \mathbf{x}^\star_{\gamma_{k-1}}(\hat{\mathbf{y}}^{k-1})$ for $k \geq 1$ in (17). This makes Algorithm 1 similar to the proximal-based decomposition algorithm in [20], which employs the proximal term $d_b(\cdot, \hat{\mathbf{x}}^\star_{k-1})$ with the Bregman distance d_b.
- *ADMM variant:* If we choose d_b to be the Euclidean distance, $\mathbf{S}$ and $\mathbf{x}_c$ such that $d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) := (1/2)\left[\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2(\mathbf{x}^\star_{\gamma_{k-1}}(\bar{\mathbf{y}}^{k-1}))_2 - \mathbf{b}\|_2^2 + \|\mathbf{A}_1(\mathbf{x}^\star_{\gamma_k}(\bar{\mathbf{y}}^k))_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2\right]$, then (1P2D) becomes a new variant of ADMM as discussed in (62).
- *Preconditioned ADMM variant:* We can choose $\mathbf{x}_c^k := (\mathbf{g}_1^k, \mathbf{g}_2^k)$, where $\mathbf{g}_1^k$ and $\mathbf{g}_2^k$ are given in (65). The step-size α_{1k} and α_{2k} can be taken as $\alpha_{1k} := \|\mathbf{A}_1\|_2^{-2}$ and $\alpha_{2k} := \|\mathbf{A}_2\|_2^{-2}$ or computed from the exact line-search rule. In this case, (1P2D) becomes a new variant of the preconditioned ADMM algorithm in [15].

In addition to the choice of $\mathbf{x}_c^k$, we can also choose an appropriate prox-function $b_{\mathscr{X}}$ for the feasible set $\mathscr{X}$ in order to define the Bregman distance d_b. For instance, if $\mathscr{X}$ is a standard simplex, i.e., $\mathscr{X} := \left\{\mathbf{x} \in \mathbb{R}^n_+ \,:\, \sum_{j=1}^n \mathbf{x}_j = 1\right\}$, then the entropy prox-function $b_{\mathscr{X}}(\mathbf{x}) := \mathbf{x}^T\ln(\mathbf{x}) + n$ becomes an appropriate choice.

7.2 Guidance on tuning the parameters

Since Algorithm 1 generates a sequence $\left\{\bar{\mathbf{w}}^k\right\}_{k\geq 0}$ that decreases the smoothed gap function $G_{\gamma_k\beta_k}(\bar{\mathbf{w}}^k)$ as required in Definition 3.2. The actual decrease on the objective residual is $f(\bar{\mathbf{x}}^k) - f^\star \leq \gamma_k(D^{\mathbf{S}}_{\mathscr{X}} - \Psi_k/\gamma_k)$. In practice, $D_k := D^{\mathbf{S}}_{\mathscr{X}} - \Psi_k/\gamma_k$ can be dramatically smaller than $D^{\mathbf{S}}_{\mathscr{X}}$ in the early iterations. This implies that increasing γ_k in the early iterations might improve practical performance.

Our strategy is based on the following observations. If γ_k increases, then τ_k also increases. Consequently, β_k decreases. Since β_k measures the primal feasibility gap $\mathscr{F}_k := \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$ due to Lemma 3.3, we only increase γ_k if the feasibility gap $\mathscr{F}_k$ is relatively high. For instance, in the case $\mathbf{x}_c^k := (\mathbf{g}_1^k, \mathbf{g}_2^k)$, we can compute the dual feasibility gap as $\mathscr{H}_k := \gamma_k\|\mathbf{A}_1^T\mathbf{A}_2((\hat{\mathbf{x}}^\star_{k+1})_2 - (\hat{\mathbf{x}}^\star_k)_2)\|$. Then, if $\mathscr{F}_k \geq s\mathscr{H}_k$ for some $s > 0$, we increase

$\gamma_{k+1} := (1 - c\tau_k)\gamma_k$ for some $c < 0$. In our implementation, we suggest the value $c = 1.05\tau_k^{-1}$ as a default option.

We can also decrease the parameter γ_k in (1P2D) by $\gamma_{k+1} := (1 - c_k\tau_k)\gamma_k$, where $c_k := d_b(\mathbf{S}\mathbf{x}^\star_{\gamma_k}(\hat{\mathbf{y}}^k), \mathbf{S}\mathbf{x}_c)/D^{\mathbf{S}}_{\mathscr{X}} \in [0,1]$ after updating the vector $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ in (1P2D) if we know a priori an upper bound estimate for $D^{\mathbf{S}}_{\mathscr{X}}$.

7.3 Parallel and distributed implementation

Suppose that f and $\mathscr{X}$ are both separable as defined in (3), where each objective component f_i and feasible set $\mathscr{X}_i$ correspond to the subsystem i ($i = 1, \ldots, p$) of a large-scale network represented by a graph as illustrated in Figure 1. The variable $\mathbf{x}_i$ represents the unknown parameters of the subsystem i, and $\mathbf{x}_i \in \mathscr{X}_i$ is its

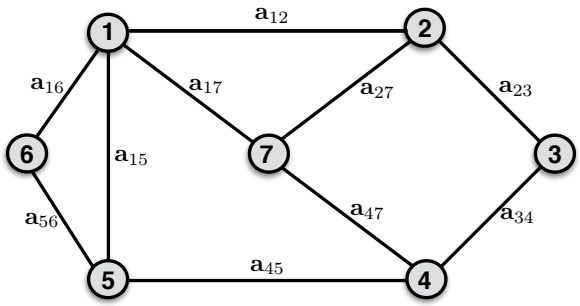

Figure 1: A graph representing the structure of problem (1) in a separable case.

local constraint. Each subsystem i communicates with its neighbors j by asking the information from them via communication links (i, j). Let $\mathbf{a}_{ij}$ be the information the subsystem i requests from its neighbor j extracted from the neighbor's variable $\mathbf{x}_j$. In this case, the information requested from all neighbors needs to be constrained by $\mathbf{b}_i$, which leads to $\sum_{j\in\mathscr{N}_i} \mathbf{a}_{ij}\mathbf{x}_j = \mathbf{b}_i$, where $\mathscr{N}_i$ denotes all the neighbors of the subsystem i, for $i = 1, \ldots, p$. We note that each subsystem i can have more than one links, the number of links leads to the number of coupling constraints. To this end, one can reformulate a convex optimization problem over this network into a constrained problem of the form (1) with separable objective, coupling constraints and separable local constraints.

Now, we assume that each $\mathscr{X}_i$ engages to a Bregman distance $d_{\mathscr{X}_i}$ with the convexity parameter $\sigma_i > 0$. We also choose either $\mathbf{S} := \|\mathbf{A}\|_2\mathbb{I}$, $\mathbf{S} := \mathrm{diag}(A_1, \cdots, A_p)$ or $\mathbf{S} := \mathrm{diag}(\|A_1\|_2\mathbb{I}_1, \ldots, \|A_p\|_2\mathbb{I}_p)$. In this case, the Bregman distance of $\mathscr{X}$ becomes $d_{\mathscr{X}}(\mathbf{x}, \mathbf{x}_c) := \sum_{i=1}^p d_{\mathscr{X}_i}(\mathbf{x}_i, \mathbf{x}_{ic})$, where the strong convexity parameter of $d_{\mathscr{X}}$ is $\sigma_d := \min_{1\le i\le p}\sigma_i$. The main step of Algorithm 1 is Step 5, where we need to perform the primal dual scheme (2P1D) or (1P2D). We show how to implement these steps in a parallel and distributed manner based on the graph structure shown in Figure 1.

Computation: The primal step in (2P1D) or (1P2D) requires to solve (17) and (34). By the separability of f and $\mathscr{X}$, (17) can be solved **in parallel**. More precisely, each subsystem i needs to estimate its local variable $\mathbf{x}_i^k$ independently by solving a subproblem of the form:

$$\mathbf{x}_i^k := \arg\min_{\mathbf{x}_i\in\mathscr{X}_i}\Big\{f_i(\mathbf{x}_i) + (\mathbf{y}^k)^T\mathbf{A}_i\mathbf{x}_i + \gamma d_{\mathscr{X}_i}(\mathbf{x}_i, \mathbf{x}_{ic}^k)\Big\},\ \ i = 1, \ldots, p,$$

where $\mathbf{y}^k$ is a local copy of the Lagrange multiplier at the iteration k for the subsystem i.

The dual step is updated as $\mathbf{y}^{k+1} := \mathbf{y}^k + \rho_k(\mathbf{A}\mathbf{x}^k - \mathbf{b})$, where $\rho_k > 0$ is a given step size. Here, each subsystem i updates its local copy of the multiplier

$$\mathbf{y}_i^{k+1} := \mathbf{y}_i^k + \rho_k\big(\sum_{j\in\mathscr{N}_i}\mathbf{a}_{ij}\mathbf{x}_{ij} - \mathbf{b}_i\big),\ \ i = 1\ldots, p,$$

and sends this sub-vector to its neighbors to compute $\mathbf{A}_i^T \mathbf{y}^{k+1}$ for the next iteration.

Communication: At each iteration k, each subsystem i requests the information from its neighbors to form the feasibility gap $\sum_{j\in\mathcal{N}_i} \mathbf{a}_{ij}\mathbf{x}_{ij} - \mathbf{b}_i$ and then updates $\mathbf{y}_i^{k+1}$. This multiplier sub-vector is then sent to the subsystem's neighbors.

Memory storage: Along with the local variable $\mathbf{x}_i$ and the feasible set $\mathbf{X}_i$, each subsystem i needs to store a copy of the dual variable $\mathbf{y}^k$ and a part of coefficient matrix $\mathbf{A}$ that represents the links to its neighbors, i.e., $\mathbf{a}_{ij}$ for $j \in \mathcal{N}_i$.

Consensus and asynchronous operation: Note that our feasibility guarantees can be used to show the "consensus" of the distributed system with a corresponding rate when the communication graph is known [12]. Intriguingly, given that algorithms are tolerant to approximate proximal operators, we might expect them to also tolerate small levels of asynchronousity. Theoretical characterization of this important variant is left for future work.

7.4 Extension to inequality constraints

The theory presented in the previous sections can be extended to solve convex optimization problems with linear inequality constraints of the form:

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^n} \{f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} \leq \mathbf{b},\, \mathbf{x} \in \mathscr{X}\}, \tag{67}$$

where f, $\mathscr{X}$, $\mathbf{A}$ and $\mathbf{b}$ are defined as in (1).

A simple way to process (67) is using a slack variable $\mathbf{s} \in \mathbb{R}^m_+$ such that $\mathbf{A}\mathbf{x}+\mathbf{s} = \mathbf{b}$ and $\mathbf{z} = (\mathbf{x},\mathbf{s})$ as the new variable. Then we can transform (67) into (1) with respect to the new variable $\mathbf{z}$.

We can also process (67) by modifying the dual steps for updating $\hat{\mathbf{y}}^k$, $\mathbf{y}^*_{\beta_k}(\bar{\mathbf{x}}^k)$ and $\bar{\mathbf{y}}^{k+1}$ in both schemes (2P1D) and (1P2D). More precisely, we update these vectors as follows:

$$\hat{\mathbf{y}}^k := \left[\beta_{k+1}^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})\right]_+,\ \mathbf{y}^*_{\beta_k}(\bar{\mathbf{x}}^k) := \left[\beta_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})\right]_+,$$

and

$$\bar{\mathbf{y}}^{k+1} = \left[\hat{\mathbf{y}}^k + (\gamma_{k+1}/\bar{L}^g)\big(\mathbf{A}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) - \mathbf{b}\big)\right]_+,$$

where $[\cdot]_+ := \max\{0,\cdot\}$. Indeed, the conclusion of Theorem 4.1 remains valid for this new variant for solving (67).

8 Numerical illustrations

In this section, we present numerical simulations on several well-studied applications from machine learning, signal and image processing, and compressive sensing. The numerical simulations are performed using MATLAB R2012b, running on a Mac OS. i7 with 2.6Ghz and 16Gb RAM. We choose the Euclidean distance $d_b(\mathbf{x},\mathbf{x}_c) := (1/2)\|\mathbf{x}-\mathbf{x}_c\|^2$ in all test cases. We terminate Algorithm 1 if both primal feasibility gap

$$\mathscr{F}_k^r := \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2/\max\{1,\|\mathbf{b}\|_2\} \leq \varepsilon_f, \ \text{ and } \ \|\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k\|_2/\max\left\{1,\|\bar{\mathbf{x}}^k\|_2\right\} \leq \varepsilon_x,$$

for given default tolerances $\varepsilon_f = 10^{-6}$ and $\varepsilon_x = 10^{-6}$ unless stated otherwise.

8.1 Actual performance vs. theoretical bounds

We demonstrate the empirical performance of the four variants of Algorithm 1 with respect to its theoretical bounds via a basic non-overlapping sparse-group basis pursuit problem:

$$\min_{\mathbf{x}\in[\mathbf{l},\mathbf{u}]\subseteq\mathbb{R}^n} \sum_{i=1}^{n_g} w_i \|\mathbf{x}_{g_i}\|_2, \quad \text{s.t. } \mathbf{A}\mathbf{x}=\mathbf{b}, \tag{68}$$

where $[\mathbf{l},\mathbf{u}]$ is a box constraint, and g_i and w_i's are the group indices and weights, respectively.

In this test, we choose $\mathbf{x}^c = \mathbf{0} \in [\mathbf{l},\mathbf{u}]$ and $d_b(\mathbf{x},\mathbf{x}_c) := (1/2)\|\mathbf{x}-\mathbf{x}^c\|^2$. We then evaluate $D_{\mathscr{X}}$ numerically, given $\mathscr{X} := [\mathbf{l},\mathbf{u}]$. We estimate $D_{\mathscr{Y}^\star}$ and $f^\star$ by solving (68) with an interior-point solver (SDPT3) [65] up to accuracy 10^{-8}. In the (2P1D) scheme, we set $\gamma_0 = \beta_0 = \sqrt{\bar{L}_g}$, while, in the (1P2D) scheme, we set $\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+1}$ with $K := 10^4$ and generate the theoretical bounds defined in Theorem 4.1.

We test the performance of the four variants using a synthetic sparse recovery problem, where $n = 1024$, $m = \lfloor n/3 \rfloor = 341$, $n_g = \lfloor n/8 \rfloor = 128$, and $\mathbf{x}^\natural$ is a $\lfloor n_g/8 \rfloor$-sparse vector. We set $\mathbf{l} := \min(\mathbf{x}^\natural)$ and $\mathbf{u} := \max(\mathbf{x}^\natural)$. Matrix $\mathbf{A}$ are generated randomly from the iid standard Gaussian distribution and $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural$. The group indices g_i is also generated randomly for $i = 1,\cdots,n_g$.

Bregman smoothing case: Figure 3 shows the empirical performance of two variants: (2P1D) and (1P2D) of Algorithm 1, where theoretical bounds are computed from Theorem 4.1. The basic algorithm refers to the

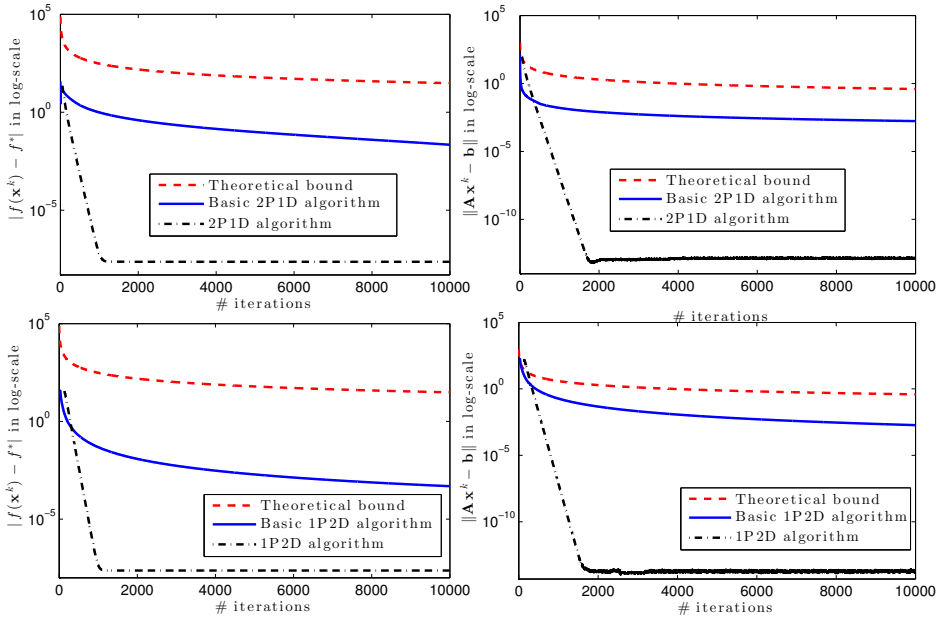

Figure 2: Actual performance vs. theoretical bounds of Algorithm 1 using Bregman smoother.

case where $\mathbf{x}_c$ is fixed and the parameters are not tuned. Hence, the iterations of the basic (1P2D) use only 1 proximal calculation and applies $\mathbf{A}$ and $\mathbf{A}^T$ once each, and the iterations of the basic (2P1D) use 2 proximal calculations and applies $\mathbf{A}$ twice and $\mathbf{A}^T$ once. In contrast, (2P1D) and (1P2D) variants whose iterations require one more application of $\mathbf{A}^T$ for adaptive parameter updates.

It is clear from Figure 3 that the empirical performance of the basic variants roughly follows the $\mathscr{O}(1/k)$ convergence rate both in terms of objective residual $|f(\bar{\mathbf{x}}^k) - f^\star|$ and the feasibility gap $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$. The deviations from the bound are due to the increasing sparsity of the iterates, which improves empirical convergence.

With a kick-factor of $c_k = -0.02/\tau_k$ and adaptive proximal-center $\mathbf{x}_c^k$ enhancements as suggested in Section 7, the tuned (2P1D) and (1P2D) variants significantly outperform theoretical predictions. Indeed, they approach the optimal solution up to 10^{-13} accuracy, i.e. $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\| \leq 10^{-13}$ after only a few hundreds of iterations.

Augmented Lagrangian smoothing case: Similarly, Figure 3 illustrates the actual performance vs. the theoretical bounds $\mathcal{O}(1/k^2)$ by using augmented Lagrangian smoothing techniques. Here, we solve the subprob-

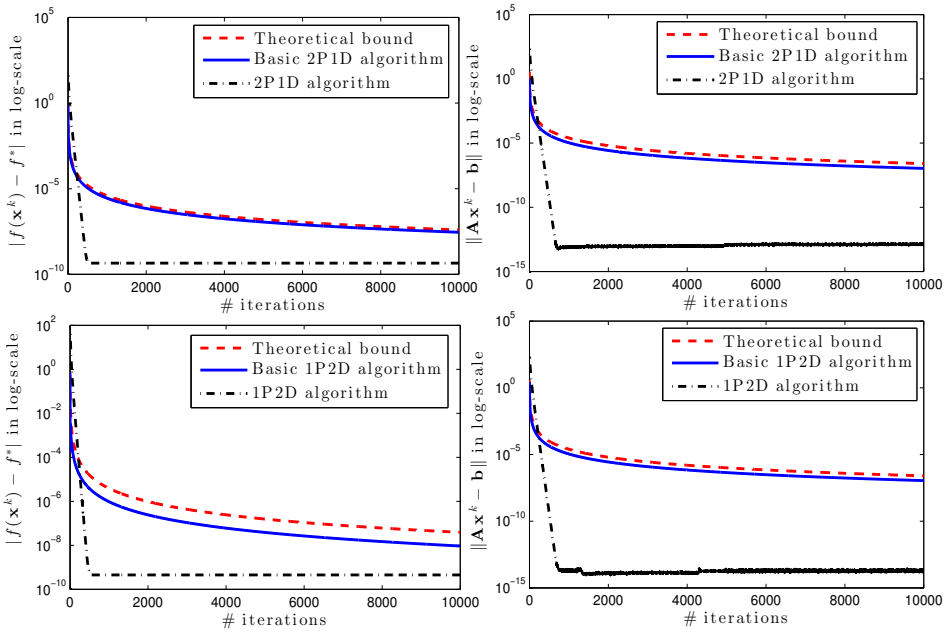

Figure 3: Actual performance vs. theoretical bounds of Algorithm 1 for augmented Lagrangian smoother.

lems (19) and (58) by using FISTA [4]. Since, we can not exactly estimate the true solution of the subproblems (19) and (58), we solve these problems up to at least the accuracy $\delta_0^2 = 10^{-8}$ as suggested by Theorem 4.1.

In this case, the theoretical bounds and the actual performance of the basis variants are very close to each other both in terms of the objective residual $|f(\bar{\mathbf{x}}^k) - f^\star|$ as well as the primal feasibility gap $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$. When the parameter γ_k is updated, the algorithms exhibit a better performance.

Strongly convex case: We demonstrate the theoretical bounds for the strongly convex case via the elastic net:

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + (\sigma/2)\|\mathbf{x}\|_2^2 \right\} \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{69}$$

where $\sigma > 0$ is a given constant, and other parameters are selected as in (68). The data of this test is also generated randomly as for (68), where $n := 2000$, $m = 700$ and $\mathbf{x}^\natural$ is 100-sparse.

We test Algorithm 1 using both (2P1D$_\sigma$) and (1P2D$_\sigma$) to solve (69) with $\sigma := 0.1$. The results are plotted in Figure 4 for both $|f(\bar{\mathbf{x}}^k) - f^\star|$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$, respectively after $K := 10^4$ iterations. The configuration of the basic variants are as before whereas the enhanced versions use a backtracking linesearch procedure to determine an approximation L_k for the Lipschitz constant L_f^g. The iterates converge better than the theoretical rate (see the appendix).

We obtain the final relative errors $(|f(\bar{\mathbf{x}}^k) - f^\star|/|f^\star|, \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2/\|\mathbf{b}\|_2)$ for both cases are $(4.0376, 2.8294) \times 10^{-6}$ and $(4.0744, 2.9064) \times 10^{-6}$, respectively. These values are $(0.8900, 0.6237) \times 10^{-6}$ and $(1.0462, 0.7400) \times 10^{-6}$, respectively, in the line-search variants, which are approximately 4 times smaller than in the basic ones.

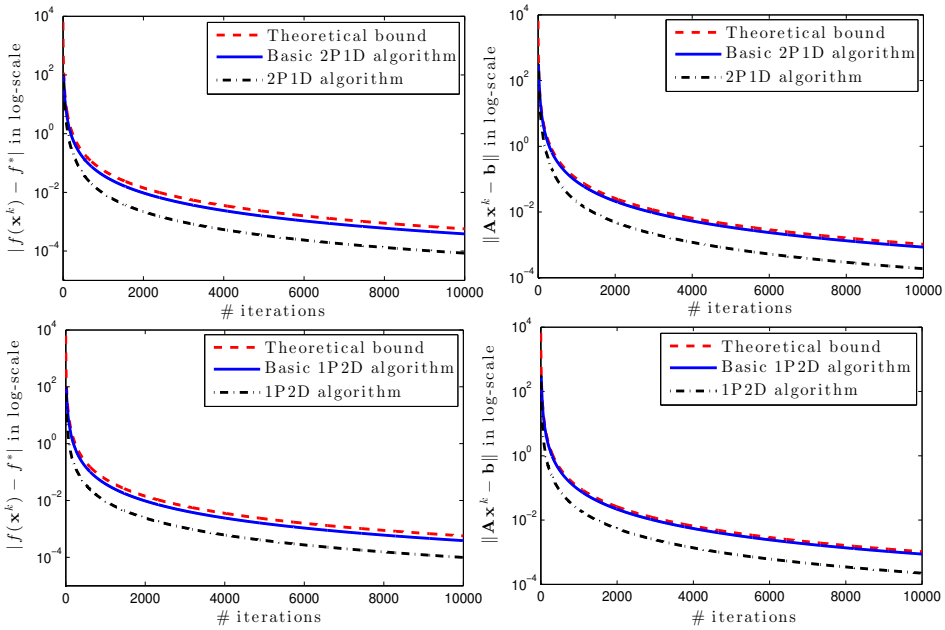

Figure 4: Actual performance vs. theoretical bounds for strongly convex case.

The relative recovery error $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2/\|\mathbf{x}^\star\|_2$ is also 3.228×10^{-7} and 3.753×10^{-7}, respectively. We also observe that after 642 (reps., 691) iterations, both algorithms reach the accuracy $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2 \leq 10^{-2}$, and after 2034 (resp., 2193) iterations, which corresponds to an approximate relative error of 10^{-3}.

We also compute the practical values of $\left\{\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2\right\}_{k\geq 0}$ and its theoretical bound shown in Corollary 5.1 for (69). The convergence of $\left\{\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2\right\}_{k\geq 0}$ and its theoretical bound is plotted in Figure 5 for both algorithms: (1P2D_σ) and (2P1D_σ), and their line-search variants, respectively.

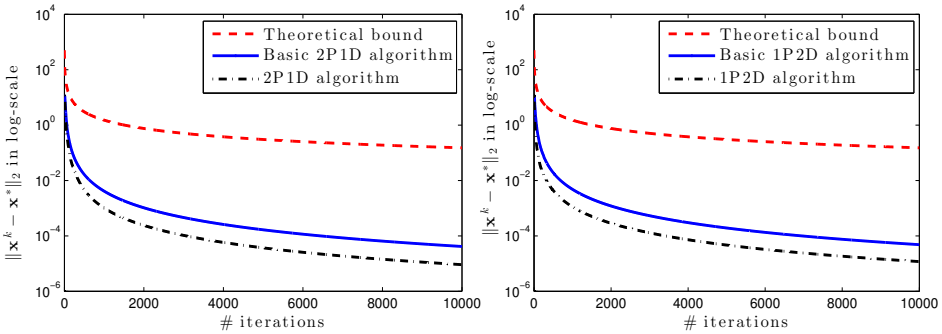

Figure 5: Actual performance vs. theoretical bound for strongly convex case (iterative sequence).

We can see that the theoretical bound given by Corollary 5.1 is far from the actual performance. This is clearly observed due to a rough estimation of the upper bound. The line-search variants takes less iterations than the basic ones, but require additional computations for the line-search procedure, which makes them in the end slower.

A new variant of preconditioned ADMM: Finally, we verify the theoretical justification of the new PADMM variant given in Corollary 6.2. The same test can be done for the new ADMM variant.

We use again the group basis pursuit problem (68) by reformulating it into the following form:

$$\min_{\mathbf{x}\in[\mathbf{l},\mathbf{u}]\subseteq\mathbb{R}^n} \sum_{i=1}^{n_g} w_i\|\mathbf{x}_{g_i}\|_2 + \delta_{\{0^m\}}(\mathbf{r}), \;\; \text{s.t. } \mathbf{A}\mathbf{x}+\mathbf{r}=\mathbf{b},\; \mathbf{r}\in[\underline{\mathbf{r}},\bar{\mathbf{r}}], \tag{70}$$

where $\delta_{\mathscr{S}}$ is the indicator function of the set $\mathscr{S}$, $\underline{\mathbf{r}}$ and $\bar{\mathbf{r}}$ is computed from the bounds $\mathbf{l}$ and $\mathbf{u}$ of $\mathbf{x}$ via the relation $\mathbf{r}=-\mathbf{A}\mathbf{x}-\mathbf{b}$.

We test the new variant of preconditioned ADMM (PADMM) and compare it with the tuned version, where we adaptively update the parameter γ_k using the strategy in Section 7. In the basis PADMM variant, we fix $\gamma_0 := 2\sqrt{2}\|\mathbf{A}\|_2/(K+1)$, where $K=10^4$ as suggested by Corollary 6.2. By using the same data as in the previous cases, we obtain the performance of this variant as shown in Figure 6.

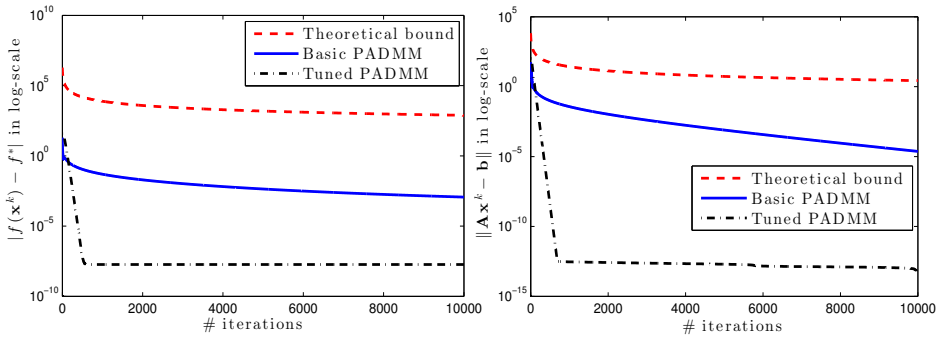

Figure 6: Actual performance vs. theoretical bound for PADMM.

As we can observe from Figure 6 that, the basis PADMM variant relatively follows the curvature of the theoretical bounds, while the tuned variant reaches very high accuracy solution after few hundreds of iterations. This behavior is similar to the (1P2D) variant using Bregman smoother tested above.

8.2 Performance robustness.

We demonstrate the performance robustness of our tuned (1P2D) variant by applying it to the following image deconvolution problem:

$$\min_{\mathbf{x}:0\le\mathbf{x}\le 255} (1/2)\|\mathscr{B}(\mathbf{x})-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_{\text{TV}}, \tag{71}$$

where $\mathbf{b}$ is a given blurry image with a known blur kernel $\mathscr{B}$, and $\|\cdot\|_{\text{TV}}$ is the isotropic total variation norm and $\lambda>0$ is a regularization parameter.

As opposed to directly using the TV-norm proximal map, we simply use the linear mapping $\mathbf{D}$ of its norm operator $\|\mathbf{x}\|_{\text{TV}} = \|\mathbf{D}\mathbf{x}\|_1$ and introduce a slack variable $\mathbf{r}=\mathbf{D}\mathbf{x}$ to split (71) into $\mathbf{x}$ and $\mathbf{r}$ variables with additional linear coupling constraint $\mathbf{r}-\mathbf{D}\mathbf{x}=0$. Hence, we can reformulate (71) into (1), where $\mathbf{r}\in\mathscr{R} := \{\hat{\mathbf{r}} \mid \hat{\mathbf{r}}=\mathbf{D}\mathbf{x}, 0\le\mathbf{x}\le 255\}$ is also bounded.

We apply the (1P2D) variant of Algorithm 1 to solve the resulting problem and compare it with the ADMM solver implemented in [16] since both algorithms have similar complexity per iteration. We choose the center point as suggested in our practical enhancement guidelines, which leads to a *new variant* of the standard ADMM method. We test two cases: without and with tuning based on our guidance. We choose the initial regularization parameters ρ_0 the same as the recent *exact* ADMM solver suggests [16].

Surprisingly, if we assume periodic boundary conditions for the TV-norm, then ADMM can efficiently obtain accurate solutions to the subproblems in computing $\mathbf{x}_1^k$ and $\mathbf{x}_2^k$. The key idea is that the operator $\mathbf{D}^T\mathbf{D} + \mathscr{B}^T\mathscr{B}$ is diagonalizable by the Fourier transform. Hence, the complexity per iteration in exact ADMM and (1P2D) is approximately the same. Note however that our algorithm does not require periodic boundary conditions to solve this class of problems, which may not be valid in other applications

Figure 7 illustrates the performance of (1P2D) and the ADMM code [16] with different values of parameter γ (resp., ρ in the ADMM solver). Our test is based on the `camera_man` image, with the regularization $\lambda = 0.01$ as done in [16]. The suggested value for ρ is $\rho = 2$ in [16]. The exact ADMM code [16] also uses a specific update rule for the penalty parameter, which is different from ours. Figure 7 shows the convergence of three algorithms wrt. three values of γ (respectively, ρ) after 100 iterations. We can see that ADMM decreases quickly first but then does not move, while (1P2D) continues to descend on the objective function.

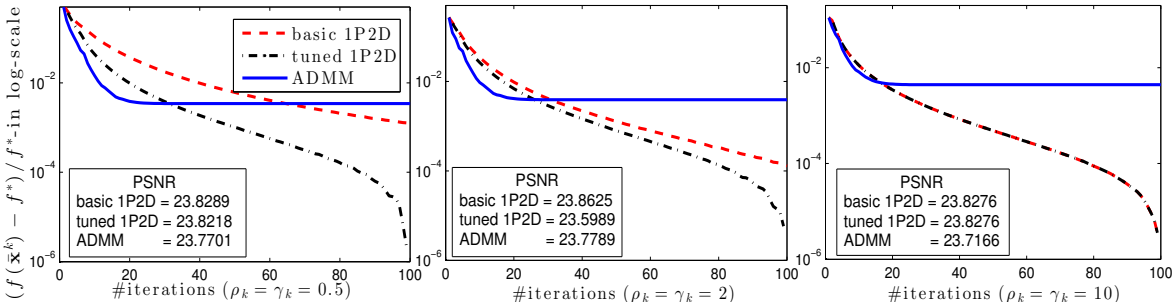

Figure 7: The performance of the augmented Lagrangian methods under different penalty parameters.

We note that the ADMM solver is sensitive to the choice of ρ. For any value of ρ, if we run up to 200 iterations then the exact ADMM algorithm diverges, which is due to their aggressive update rule on the penalty parameter.

8.3 Inexact computations.

In this test, we study the empirical impact of inexact proximal operator calculations to the performance of Algorithm 1. Again, we choose the (1P2D) variant, which has similar complexity per iteration as preconditioned ADMM [15]. For this, we use a Schatten norm based regularizer on a Poisson likelihood data model:

$$\min_{\mathbf{x}\in\mathscr{X}} (\mathscr{B}(\mathbf{x}))^T\mathbf{1} - \sum_{i=1}^{m} \mathbf{c}_i \log((\mathscr{B}(\mathbf{x}))_i + \mathbf{b}) + \lambda\|\mathbf{x}\|_S, \tag{72}$$

where $\mathscr{X} := \mathbb{R}^n_+$, $\mathbf{c}$ is a given photon count vector in $\mathbb{Z}^m$, $\mathbf{b}$ is the background intensity, $\lambda > 0$ is a chosen regularization parameter, and $\mathscr{B}$ is a blur kernel. This likelihood model is quite common in scientific imaging problems.

The work in [40] proposed a norm based on exploiting self-similarities within the images via $\|\mathbf{x}\|_S := \|\mathrm{mat}(\mathscr{H}(\mathbf{x}))\|_\star$, which is the Schatten-norm of a matrix $\mathrm{mat}(\mathscr{H}(\mathbf{x}))$ for a suitably chosen linear operator $\mathscr{H}$. Since the proximal operator regarding the second term $f_2(\mathbf{x}) := \lambda\|\mathbf{x}\|_S + \delta_{\mathscr{X}}(\mathbf{x})$, where $\delta_{\mathscr{X}}$ is the indicator of $\mathscr{X}$, does not have a closed form, we need to iteratively compute it.

The resulting inexact computation affects the performance of optimization algorithms. Here, we compare our new PADMM variant of Algorithm 1 (called tuned 1P2D) with PADMM and PADMM based on our tuning strategy in the enhancement paragraph as well as the exact ADMM solver provided by [40]. Here, the ADMM solver exploits boundary conditions and Fourier transform to invert $I + \mathscr{B}^T\mathscr{B}$ for solving its subproblems.

When $\mathbf{b}$ is zero (i.e., there is no background), then the logarithmic term pose computational problems since its gradient is no longer Lipschitz. Fortunately, the proximal operator of the log function can be efficiently calculated.

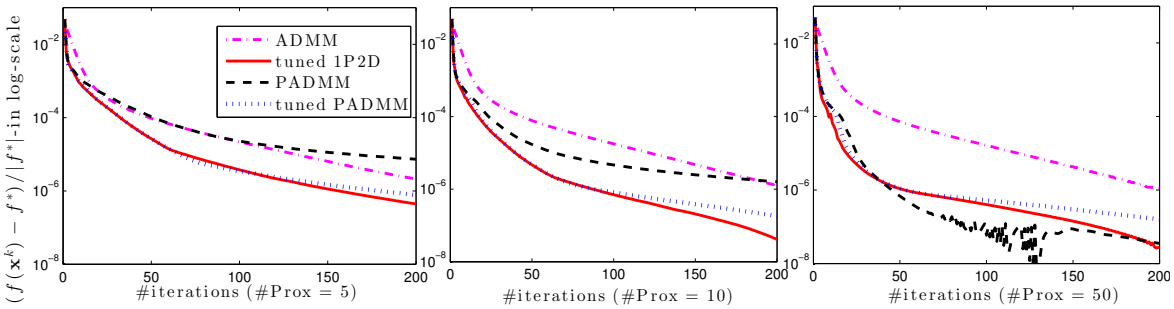

Figure 8: The performance of four algorithms on the Clown image [40].

We test these algorithms on the Clown image where we take the regularization parameter $\lambda = 0.055$ suggested in [40]. We use the `Denoise` solver in [40] to approximately compute the prox-operator of f_2 with inner iterations nProx$= 5, 10, 50$, where we can warm start each iteration using each algorithms current estimate. The exact ADMM solver is already implemented with penalty parameter updates.

Figure 8 illustrates that our tuned (1P2D) solver and PADMM are quite robust to inexact prox calculations and outperform exact ADMM for a range of nProx values. Against intuition, we observe that PADMM exhibits numerical instability when nProx is highest. Overall, our algorithm provides the best time to reach an ε-solution since doubling nProx roughly doubles the overall time. For instance, nProx $= 5$ and 200 iterations roughly takes the same time as nProx $= 10$ and 100 iterations, where our algorithm provides the best accuracy.

In this setting, our solver and PADMM do not require periodic boundary conditions. When this assumption is removed, the subproblem are no longer dominated by just prox calculations. Then, we expect our algorithm obtain better timing performance due to its parallel updates.

8.4 Additional comparisons with state-of-the-art.

We compare our algorithms with existing state-of-the-art Matlab codes for solving five well-studied problems: standard basis pursuit, group-sparse basis pursuit, robust PCA, square-root LASSO and support vector machines with the Hinge loss. While there are several software packages that can be used to solve these problems, we only select few of representatives which we find as the most efficient methods for corresponding problems.

8.4.1 Standard basis pursuit.

We consider the standard basis pursuit problem arising from compressive sensing [25]:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{73}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$.

In this example, we compare our algorithms with YALL1 [71] and SPGL$_1$ [8] which are well-known solvers for the basis pursuit problem. We use the data from the benchmark collection `Sparco` [9]. For YALL1 and SPGL$_1$, we use the default settings and all the algorithms are terminated with the accuracy 10^{-6}. Within our methods, we run three algorithms: (1P2D) via Bregman distance smoothing, inexact 1P2D(1) (with only one FISTA iteration) and inexact 1P2D(5) (with 5 FISTA iterations) via augmented Lagrangian smoothing. The

two last algorithms are inexact variants of Algorithm 1 using the augmented Lagrangian smoother. Table 3 shows the problems selected from the `Sparco` test collection [8] that we use for our test.

Table 3: The `Sparco` test problems used

Problems	ID	m	n	$\|\mathbf{b}\|_2$	Operators
`gcosspike`	5	300	2048	8.1e+1	Gaussian ensemble, DCT
`p3poly`	6	600	2560	2.2e+0	Gaussian ensemble, wavelet
`sgnspike`	7	600	2560	2.2e+0	Gaussian ensemble
`zsgnspike`	8	600	2560	2.9e+0	Gaussian ensemble
`gausspike`	11	256	1024	8.7e+1	Gaussian ensemble
`srcsep1`	401	29166	57344	2.2e+1	windowed DCT
`srcsep2`	402	29166	86016	2.3e+1	windowed DCT
`phantom1`	501	629	4096	1.1e+1	restricted FPT, wavelet
`blurrycam`	701	65536	65536	1.3e+2	blurring, wavelet
`blurspike`	702	16384	16384	2.2e+0	blurring

The numerical results and performance information are reported in Table 4 for 10 problems from Table 3. Our algorithms and YALL1 are still superior to SPGL_1 both in terms of number of iterations, matrix-vector multiplications and CPU time, while producing very similar final objective value $f(\mathbf{x}^k)$ and the feasibility gap $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|_2$. YALL1 performs quite well compared to our methods in terms of timing. However, it fails for the last two problems (i.e., `blurrycam` and `bluspike`) due to their parameter update rules.

We also note that within $s := 1$ to 5 FISTA iterations, the inexact (1P1D(s)) algorithms still perform well and produce more accurate solutions when the inner iteration number is increasing.

8.4.2 Sparse-group basis pursuit.

We consider again the sparse-group basis pursuit problem (68). In this case, we compare our algorithms and the group YALL1 solver [71], which we find one of the most efficient algorithm for solving (68). A further comparison with SPGL_1 can be found in [71].

One of the most common ways to compare the performance of different algorithms is using performance profile concept [24]. In this example, we benchmark seven algorithms with performance profiles.

Recall that a performance profile is built based on a set $\mathscr{S}$ of n_s algorithms (solvers) and a collection $\mathscr{P}$ of n_p problems. Suppose that we build a profile based on computational time (but the same concept can be used for different measurements). We denote by

$$T_{p,s} := \textit{computational time required to solve problem } p \textit{ by solver } s.$$

We compare the performance of algorithm s on problem p with the best performance of any algorithm on this problem. That is, we compute the performance ratio $r_{p,s} := \frac{T_{p,s}}{\min\{T_{p,\hat{s}} \mid \hat{s} \in \mathscr{S}\}}$. Now, let

$$\tilde{\rho}_s(\tilde{\tau}) := (1/n_p)\text{size}\left\{p \in \mathscr{P} \mid r_{p,s} \leq \tilde{\tau}\right\} \text{ for } \tilde{\tau} \in \mathbb{R}_+.$$

The function $\tilde{\rho}_s : \mathbb{R} \to [0,1]$ is the probability for solver s that a performance ratio is within a factor $\tilde{\tau}$ of the best possible ratio. We use the term "performance profile" for the distribution function $\tilde{\rho}_s$ of a performance metric. We plotted the performance profiles in log-scale, i.e.

$$\rho_s(\tau) := (1/n_p)\text{size}\left\{p \in \mathscr{P} \mid \log_2(r_{p,s}) \leq \tau := \log_2 \tilde{\tau}\right\}.$$

Table 4: Comparison of the five algorithms: (1P2D), 1P2D(1), 1P2D(5), YALL1 and SPGL$_1$.

	1P2D	1P2D(1)	1P2D(5)	YALL1	SPGL$_1$	1P2D	1P2D(1)	1P2D(5)	YALL1	SPGL$_1$
Problems	#Iterations					CPU time [s]				
gcosspike	330	275	274	**208**	1026	0.87	0.74	2.16	**0.62**	2.53
p3poly	306	100	**98**	252	1775	14.32	**4.81**	16.11	13.34	67.10
sgnspike	346	157	**156**	178	291	0.96	**0.50**	1.48	0.61	1.01
zsgnspike	331	307	307	**152**	320	1.59	1.53	4.65	**0.91**	1.87
gausspike	368	320	**319**	170	516	0.31	0.29	0.67	**0.19**	0.52
srcsep1	380	331	**330**	426	1580	22.65	**19.44**	67.88	36.95	119.60
srcsep2	376	326	**325**	334	1310	34.64	**29.73**	102.88	58.19	155.12
phantom1	291	285	285	**166**	712	1.16	1.02	2.70	**0.50**	2.42
blurrycam	1042	3496	**569**	failed	3629	**23.48**	72.97	39.08	failed	152.96
blurspike	1255	4191	**797**	failed	2159	**5.83**	18.86	10.14	failed	17.16
Problems	#$\mathbf{Ax}$					#$\mathbf{A}^T\mathbf{y}$				
gcosspike	332	552	1600	**312**	1815	331	**276**	1325	416	1028
p3poly	308	**202**	590	378	3279	307	**101**	491	504	1777
sgnspike	348	316	902	**267**	482	347	**158**	745	178	293
zsgnspike	333	616	1766	**228**	557	332	308	1458	**152**	322
gausspike	370	642	1840	**255**	858	369	321	1520	340	518
srcsep1	**382**	664	1962	639	2639	**381**	332	1631	852	1582
srcsep2	**378**	654	1922	501	2122	377	**327**	1596	668	1312
phantom1	293	572	1687	**249**	1014	292	286	1401	**166**	599
blurrycam	**1044**	6994	3420	failed	6800	**1043**	3497	2850	failed	3631
blurspike	**1257**	8384	4180	failed	4127	**1256**	4192	3382	failed	2161
Problems	The objective value $f(\mathbf{x}^k)$					$\|\mathbf{Ax}^k-\mathbf{b}\|_2/\|\mathbf{b}\|_2\times 10^5$				
gcosspike	181.484	183.050	**181.481**	181.483	181.482	0.096	**0.087**	0.091	3.479	0.187
p3poly	1748.023	1838.254	1747.982	**1747.954**	1748.363	0.079	0.079	0.082	1.374	**0.001**
sgnspike	20.620	20.620	**20.619**	20.621	20.620	0.211	0.090	**0.090**	1.324	9.963
zsgnspike	28.927	28.927	**28.927**	28.928	28.927	0.349	0.093	**0.092**	1.598	7.169
gausspike	24.041	24.041	**24.041**	24.041	24.041	0.152	0.093	**0.092**	1.628	0.112
srcsep1	1057.583	1059.361	**1057.228**	1057.974	1058.821	0.123	0.093	**0.091**	0.954	0.647
srcsep2	1093.134	1094.450	**1092.807**	1097.060	1093.961	0.118	0.092	**0.094**	1.050	0.446
phantom1	202.697	202.828	**202.696**	202.856	202.783	0.572	0.085	**0.085**	1.148	2.412
blurrycam	**10276.681**	10276.682	10276.691	failed	10276.717	0.125	0.099	0.097	failed	**0.075**
blurspike	576.482	576.482	576.482	failed	**576.474**	0.125	0.100	**0.100**	failed	9.067

The data of this test is generated as follows. The problem size is set to $n := s \times 5120$, $m := \lfloor n/3 \rfloor$ and $n_g := \lfloor m/4 \rfloor$ for $s := 1, \cdots, 20$. Matrix $\mathbf{A}$ is drawn randomly from standard Gaussian distribution with 50% correlated columns. Vector $\mathbf{b} := \mathbf{Ax}^\star + \sigma$, where $\mathbf{x}^\star$ is a given test vector generated also randomly with the standard Gaussian distribution, and σ is a Gaussian noise.

Figure 9 shows the performance profile of 7 algorithms: 6 variants of Algorithm 1 and group_YALL1 [71] in terms of iteration numbers, computational time (in second), the number of nonzero groups and the relative recovery errors $\|\mathbf{x}^k - \mathbf{x}^\star\|_2/\|\mathbf{x}^\star\|_2$. These performance profiles are built from 37 problems for size $[m,n,n_g] = [1706, 5120, 427]$ to $[8533, 25600, 2133]$ without additive Gaussian noise. The y-axis of these figures shows the problem ratio $\rho_s(\tau)$. If the problem ratio $\rho_s(\tau)$ is closer to 1, then the corresponding algorithm has a better performance. The x-axis shows how many times (2^τ) one algorithm is better than the others in $\log_2$-scale.

We can observe from the performance profiles in Figure 9 for the noiseless case that: The (1P2D) variant is

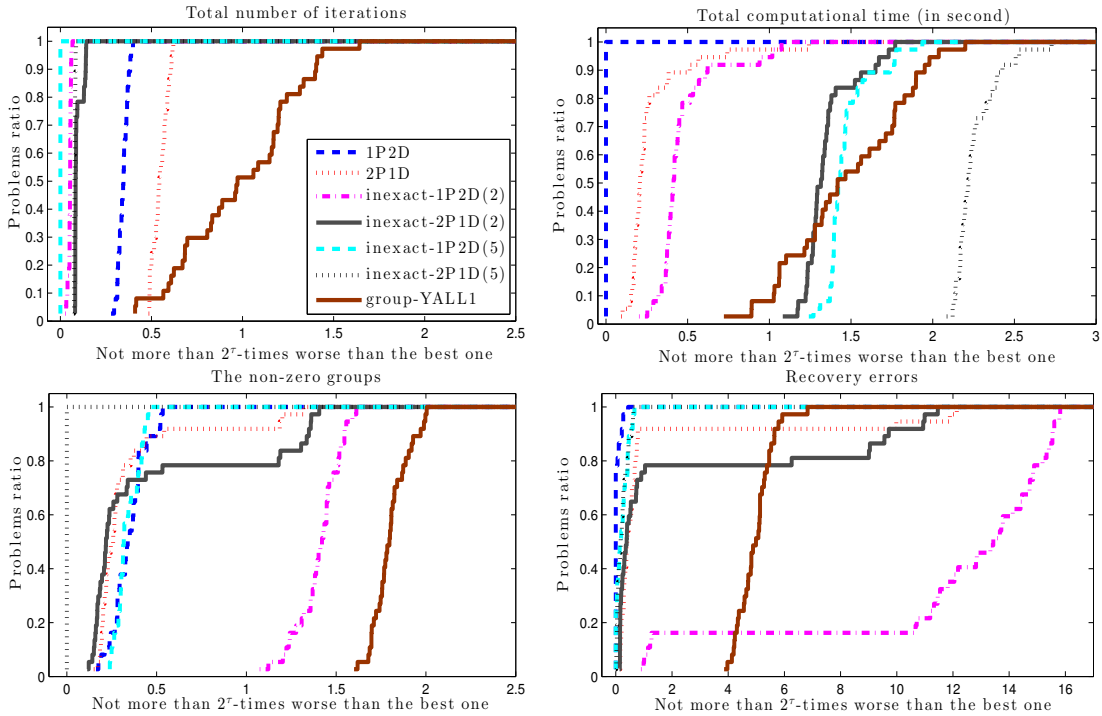

Figure 9: The performance profiles of 7 algorithms on synthetic data without noise

the best one in terms of computational time while produces relatively good results (number of nonzero groups, solution recovery errors) compared to the rest. The inexact (2P1D) variant with 5 FISTA iterations gives the best results (number of nonzero groups, solution recovery errors) but is slow due to two primal steps. While the computational time of our algorithms slightly increases with respect to the problem size, it increases linearly in group_YALL1 due to the solution of linear systems. The inexact (2P1D) is more robust to the FISTA iterations than the inexact (1P2D) one.

Figure 10 presents the performance profiles when we add 5% Gaussian noise to the model. The performance of our algorithms basically remains the same as in the noiseless case, while the number of nonzero groups in group_YALL1 is increasing significantly compared to ours. If we increase the noise level up to 10%, group_YALL1 starts oscillating and cannot converge to the solution with the desired accuracy. This happens due to the effect of the fixed penalty parameter in group_YALL1. We note that if we update this parameter, the linear system in group_YALL1 needs to be resolved, which slows down significantly the performance of the algorithm except some tricks are exploited.

8.4.3 Robust principle component analysis.

We consider the following robust principle component analysis (RPCA) problem:

$$\min_{\mathbf{X},\mathbf{Y}\in\mathbb{R}^{m\times n}} \{\lambda\|\mathrm{vec}(\mathbf{X})\|_1 + \|\mathbf{Y}\|_\star : \mathbf{X}+\mathbf{Y}=\mathbf{M}\}, \tag{74}$$

where $\mathbf{M}\in\mathbb{R}^{m\times n}$ is a given matrix, $\|\cdot\|_\star$ is the nuclear norm and $\lambda>0$ is a regularization parameter. As suggested in [14], we can choose $\lambda := \frac{c}{\sqrt{m}}$ to get a perfect recovery (i.e., with high probability), where $c>0$ is a scaling constant.

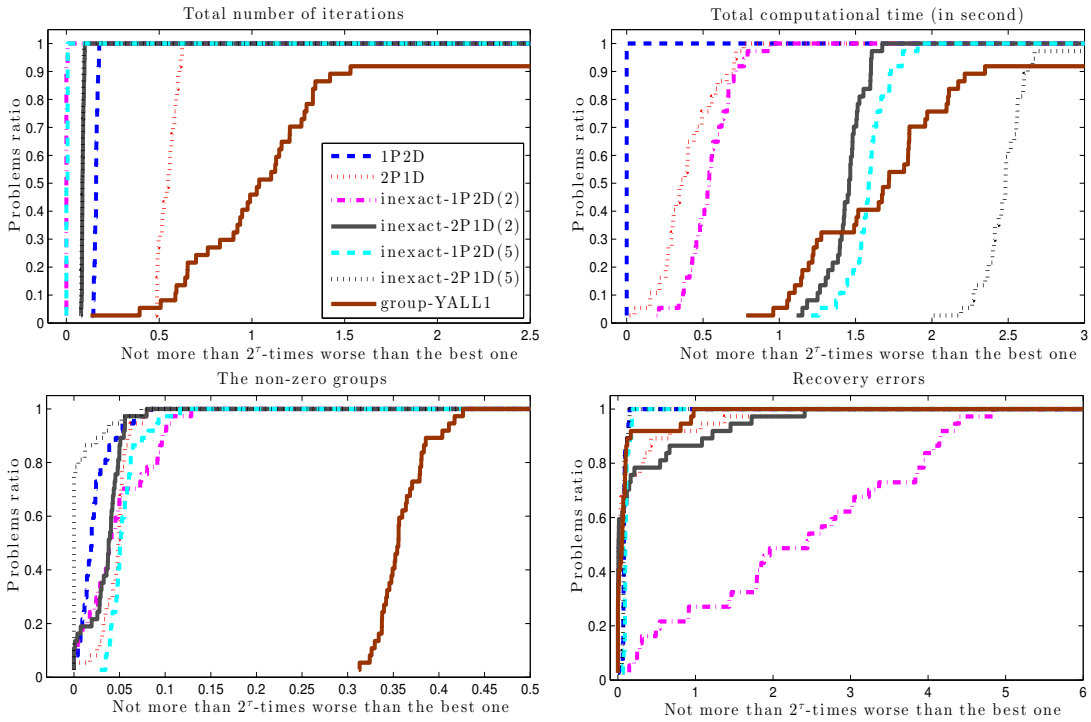

Figure 10: The performance profiles of 7 algorithms on synthetic data with 5% Gaussian noise

In this example, we demonstrate our (1P2D) algorithm on the video clip taken from a surveillance camera in a subway station, which is available at `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`. We crop 200 gray frames from this video clip and preprocess it to obtain a $20'800 \times 200$ matrix as an input $\mathbf{M}$. By tuning the regularization parameter λ, we pick the best possible value $\lambda := 0.01$. We run our (1P2D) algorithm and compare it with three other open-source codes: exact ADMM, inexact ADMM [41] and TFOCS [6]. All the algorithms are terminated with the same accuracy 10^{-3}.

The results and performance of these algorithms are reported in Table 5, where #svd is the number of SVDs required by the algorithms, $F(\mathbf{X}^k, \mathbf{Y}^k) := \lambda \|\mathrm{vec}(\mathbf{X})\|_1 + \|\mathbf{Y}\|_\star$. We can see from Table 5, (1P2D)

Table 5: The results and performance of four algorithms on the real-world data

Algorithms	#iterations	#svd	$F(\mathbf{X}^k, \mathbf{Y}^k)$	$\frac{\|\mathbf{X}^k+\mathbf{Y}^k-\mathbf{M}\|_F}{\|\mathbf{M}\|_F}$	Time[s]
(1P2D)	13	14	547845.12485	0.0004029	10.53
exactADMM	4	662	548333.09286	0.0000676	458.75
inexactADMM	19	19	548551.75715	0.0004988	9.33
TFOCS	38	122	566257.63794	0.0008508	111.89

requires fewest SVD operations and has similar computational time as inexact ADMM, while reaches a better objective value $F(\mathbf{X}^k, \mathbf{Y}^k)$ and the relative feasibility gap. The exact ADMM produces a better solution in terms of quality (lower relative feasibility gap) but requires too many SVDs.

The frame 25 of this video is plotted in Figure 11, which illustrates how the output of the algorithms can be

presented in object separation context. We can see from this plot that the objects (humans) can be considered

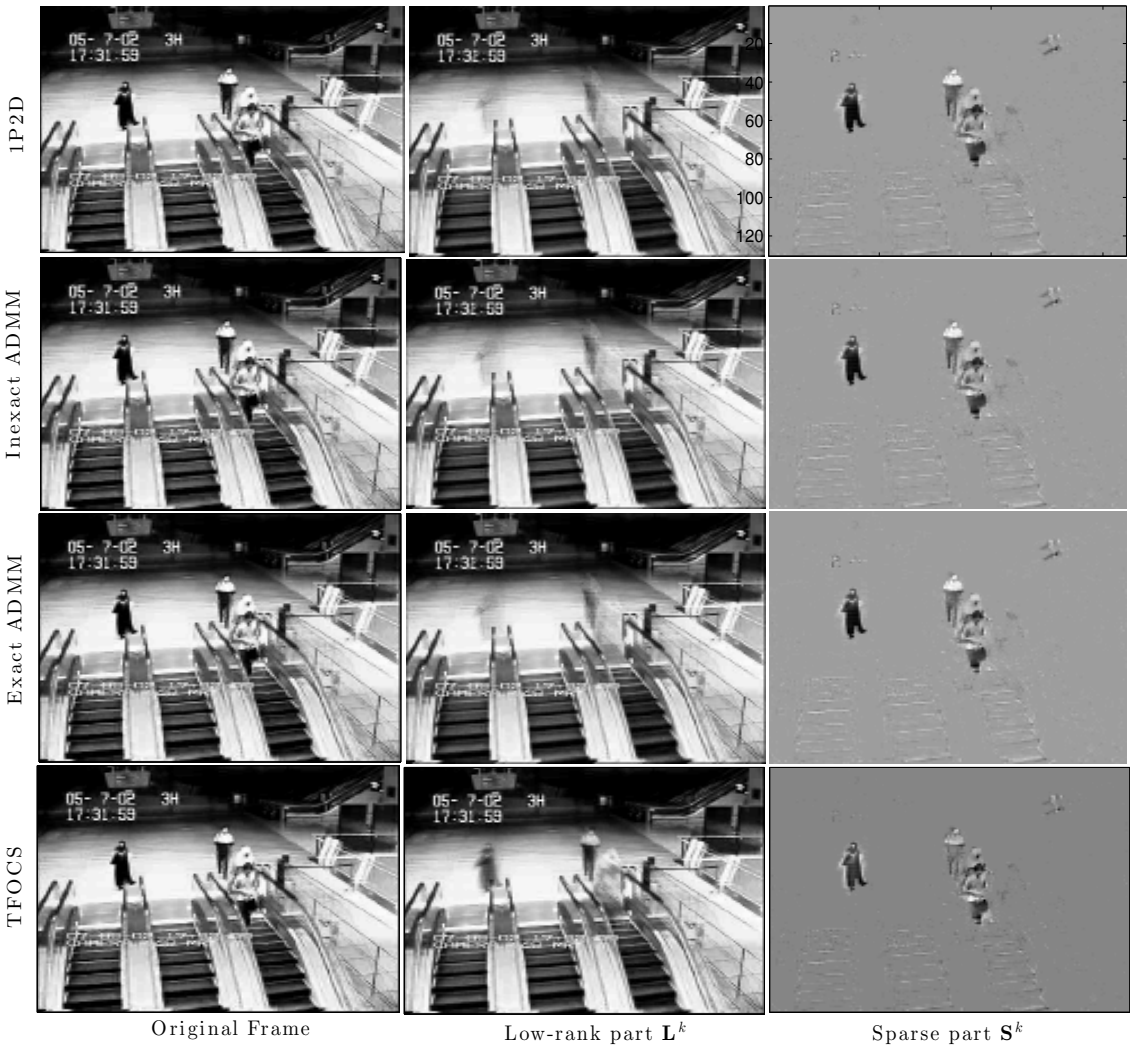

Figure 11: The results of four algorithms on the frame 25 of the video clip

as sparse representation and are separated from the background. As can be observed from the second column in Figure 11, (1P2D) and ADMMs give a better low-rank image estimate as compared to TFOCS.

8.4.4 Square-root LASSO.

Since the (1P2D) variant of Algorithm 1 has similar cost-per-iteration as ADMM, we compare this algorithm with the state-of-the-art solvers such as TFOCS, ADMM and PADMM.

For this purpose, we choose the square-root LASSO problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2+\lambda\|\mathbf{x}\|_1, \tag{75}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ are given and $\lambda > 0$ is a regularization term. By introducing a new variable $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$, (75) can be reformulated in the form of (1):

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{r}\in\mathbb{R}^m} \lambda \|\mathbf{x}\|_1 + \|\mathbf{r}\|_2, \text{ s.t. } \mathbf{A}\mathbf{x} - \mathbf{b} - \mathbf{r} = 0. \tag{76}$$

As shown in [7] that the regularization parameter λ can be set at $\lambda = c\Phi^{-1}(1 - 0.5\alpha/n)$ for given $c > 1$ and $\alpha \in (0,1)$. The suggested values for c and α are 1.1 and 0.05, respectively. By choosing this value of λ, we can probably recover $\mathbf{x}$ with probability $1 - \alpha = 0.95$.

We mimic the basis pursuit problem before and generate 5 problems of size $(m,n,s) = i(350,1000,100)$, where $i = 1,\ldots,5$ and s is the sparsity. We generate the matrix $\mathbf{A}$ randomly from Gaussian distribution with 0.5 correlated columns. Vector $\mathbf{b}$ is generated as $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathbf{n}$, and $\mathbf{n}$ is Gaussian noise with distribution $\mathcal{N}(0, 0.1)$.

We tune all the augmented Lagrangian algorithms: (1P2D), the preconditioning ADMM (PADMM) and the exact ADMM (ADMM). In these algorithms, we use the same strategy to tune the smoothness parameter γ_k and the penalty parameter ρ_k, as we observe this works best for three algorithms. The center point $\mathbf{x}_c^k$ in Algorithm 1 is chosen as discussed in the enhancement paragraph. In stark contrast to the ADMM and PADMM, our subproblems with respect to $\mathbf{x}$ and $\mathbf{r}$ are solved in **parallel**. Note that the ADMM requires one matrix inversion $\mathbf{I} + \mathbf{A}^T\mathbf{A}$.

A Monte Carlo run of size 10 shows that our algorithm is not only more accurate but is also faster (cf., Table 6). We count the number of matrix-vector multiplications both in $\mathbf{A}\mathbf{x}$ and $\mathbf{A}^T\mathbf{x}$ since these are more expensive than the prox operators. Since the iterative vector $\mathbf{x}$ is sparse, the multiplication $\mathbf{A}^T\mathbf{y}$ is more expensive. As we can see through this example that ADMM requires more iterations than Algorithm 1 and PADMM while produces lower accurate solutions. At the same time, TFOCS is slowest and least accurate while sometimes obtaining better estimation error.

8.4.5 Binary linear support vector machine.

This example is concerned the following binary linear support vector machine problem of the Hinge loss function:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ F(\mathbf{x}) := \sum_{j=1}^m \ell_j(y_j, \mathbf{w}_j^T\mathbf{x} - \mathbf{b}_j) + g(\mathbf{x}) \Big\}, \tag{77}$$

where $\ell_j(s,\tau)$ is the Hinge loss function given by $\ell_j(s,\tau) := \max\{0, 1 - s\tau\} = [1 - s\tau]_+$, $\mathbf{w}_j$ is the column of a given matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^n$ is the bias vector, $\mathbf{y} \in \{-1,+1\}^m$ is a classifier vector g is a given regularization function, e.g., $g(\mathbf{x}) := \frac{\lambda}{2}\|\mathbf{x}\|_2^2$ for ℓ_2-regularizer or $g(\mathbf{x}) := \lambda\|\mathbf{x}\|_1$ for ℓ_1-regularizer ($\lambda > 0$ is a regularization parameter).

By introducing a slack variable $\mathbf{r} = \mathbf{W}\mathbf{x} - \mathbf{b}$, we can write (77) in terms of (1) as

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n, \mathbf{r}\in\mathbb{R}^m} & \Big\{ \sum_{j=1}^m \ell_j(y_j, \mathbf{r}_j) + g(\mathbf{x}) \Big\} \\ \text{s.t.} & \mathbf{W}\mathbf{x} - \mathbf{r} = \mathbf{b}, \end{array} \tag{78}$$

Now, we can apply the (1P2D) variant to solve this resulting problem. We test this algorithm on (78) and compare it with LibSVM [19]. We select only two problems from the LibSVM data set available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/` for our test. The first problem is `a1a`, which has size $p = 119$ features and $N = 1605$ data points, while the second problem is `news20`, which has size $p = 1355191$ features and $N = 19996$ data points.

We compare two algorithms in terms of the final objective values $F(\mathbf{x}^k)$, the classification accuracy $\text{ca}_\lambda := 1 - N^{-1}\sum_{j=1}^N \big[\text{sign}(\mathbf{W}\mathbf{x}^k - \mathbf{r}) \neq \mathbf{y})\big]$ and the computational time. The results of our test are reported in Table 7.

As can be seen from these results that both solvers give relatively the same objective values, the accuracy for these two problems, while the computational of (1P2D) is much lower than `LibSVM`. We note that

Table 6: Performance comparison of Algorithm 1, PADMM, ADMM, and TFOCS.

Size			# Iterations			
m	n	s	(1P2D)	PADMM	ADMM	TFOCS
350	1000	100	1331	1592	3665	5000
700	2000	200	1311	1398	2861	5000
1050	3000	300	1307	1335	2797	5000
1400	4000	400	1318	1330	2631	5000
1750	5000	500	1316	1322	2594	5000
Size			$\#\mathbf{A}\mathbf{x}/\#\mathbf{A}^T\mathbf{y}$			
m	n	s	(1P2D)	PADMM	ADMM	TFOCS
350	1000	100	1332/ 2661	1593/ 3184	3666/ 7330	15996/ 5523
700	2000	200	1312/ 2621	1399/ 2796	2862/ 5720	16005/ 5548
1050	3000	300	1308/ 2613	1336/ 2670	2798/ 5593	15989/ 5826
1400	4000	400	1319/ 2635	1331/ 2659	2632/ 5260	16018/ 5801
1750	5000	500	1317/ 2630	1323/ 2644	2595/ 5187	16022/ 5790
Size			Objective values $f(\bar{\mathbf{x}}^k)$			
m	n	s	(1P2D)	PADMM	ADMM	TFOCS
350	1000	100	31.424461	31.424537	31.424762	32.652869
700	2000	200	74.917422	74.917552	74.919787	77.039976
1050	3000	300	120.904351	120.904523	120.909089	123.684820
1400	4000	400	150.458042	150.458275	150.465146	156.510366
1750	5000	500	192.030170	192.030441	192.040217	201.906842
Size			Recovery errors $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|/\|\mathbf{x}^\star\|$			
m	n	s	(1P2D)	PDMM	ADMM	TFOCS
350	1000	100	0.15120	0.15122	0.15180	0.14713
700	2000	200	0.04689	0.04689	0.04707	0.04447
1050	3000	300	0.03165	0.03166	0.03181	0.02947
1400	4000	400	0.03013	0.03014	0.03025	0.04040
1750	5000	500	0.03802	0.03803	0.03824	0.04973

`LibSVM` was implemented in C++ while (1P2D) is simply a Matlab code. `LibSVM` becomes slower when the parameter λ getting smaller due to the active-set strategy. The (1P2D) algorithm is almost independent of the regularization parameter λ, which is different from active-set methods. In addition, the performance of (1P2D) can be improved by taking account its parallelization ability, which has not been exploited yet in our Matlab implementation.

To immediately see the performance without looking at the numbers in Table 7, we plot the results in Figures 12 and 13 for two separate problems, respectively.

9 Conclusions

We introduce a model-based excessive gap (MEG) technique for constructing and analyzing first-order methods that numerically approximate an optimal solution of (1). Thanks to a combination of smoothing strategies and MEG, we introduce, to the best of our knowledge, the first algorithmic schemes for (1) that theoretically obtain optimal convergence rates directly without averaging the iterates and that seamlessly handle the p-decomposability structure. Surprisingly, our analysis techniques enable inexact characterizations, which is important for the augmented Lagrangian versions with lower-iteration counts. We expect a deeper under-

Table 7: The results of two algorithms on two real-world data problems

Problem	The parameter values									
λ^{-1}	10^{-3}	111.1	222.2	333.3	444.4	555.6	666.7	777.8	888.9	10^{3}
The accuracy of problem `a1a`										
(1P2D)	0.7539	0.8717	0.8717	0.8710	0.8710	0.8710	0.8710	0.8710	0.8710	0.8710
`LibSVM`	0.7539	0.8692	0.8698	0.8698	0.8698	0.8698	0.8698	0.8698	0.8679	0.8698
The CPU time [in second] of problem `a1a`										
(1P2D)	4.4045	4.3769	4.4246	4.4941	4.6238	4.5175	4.4836	4.4719	4.7179	4.8097
`LibSVM`	0.2549	2.1909	4.3884	5.8583	8.3662	11.2350	11.7036	12.9832	17.1424	17.4362
The accuracy of problem `news20`										
(1P2D)	0.5001	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987
`LibSVM`	0.5001	0.9987	0.9987	0.9987	0.9987	0.9988	0.9988	0.9988	0.9988	0.9988
The CPU time [in second] of problem `news20`										
(1P2D)	762.31	1023.22	994.64	1043.06	984.24	989.70	1064.33	1073.94	984.47	1018.35
`LibSVM`	890.26	1440.28	1449.23	1439.77	1434.27	1518.56	1560.38	1557.48	1535.19	1530.71

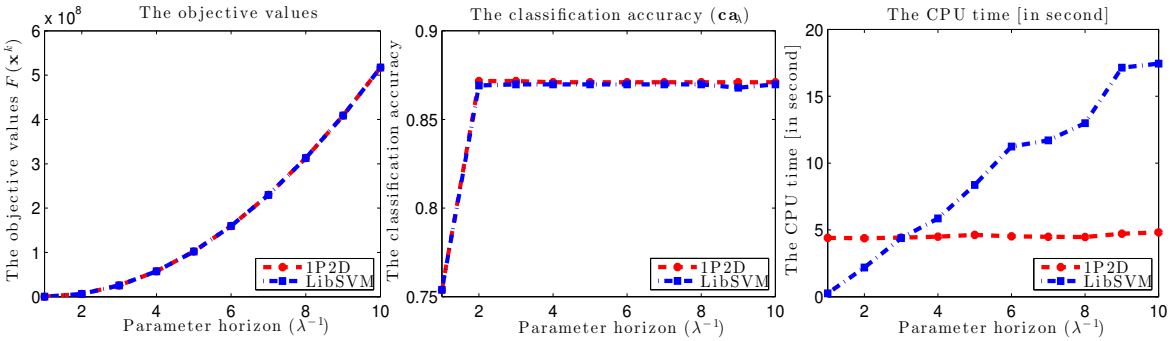

Figure 12: The results of two algorithms on the small `a1a` problem

standing of MEG and different smoothing strategies to help us in tailoring adaptive update strategies for our schemes (as well as several other connected and well-known schemes) in order to further improve the empirical performance.

Acknowledgments

This work is supported in part by the European Commission under the grants MIRG-268398 and ERC Future Proof, and by the Swiss Science Foundation under the grants SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633.

A The proofs of technical statements

This appendix provides the technical proofs of Lemmas and Theorems introduced in the main text.

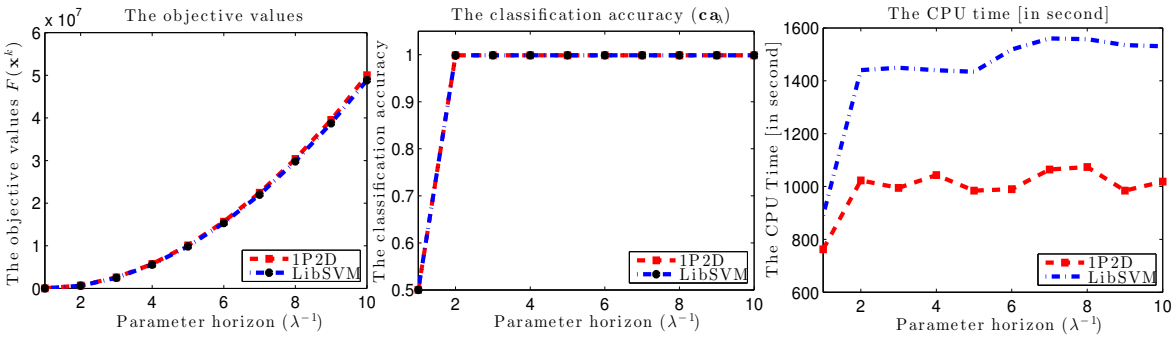

Figure 13: The results of two algorithms on the large-scale `news20` problem

A.1 The proof of Lemma 3.3: Bounds on the objective residual and feasibility gap.

By induction, it follows from Definition 3.2 that $G_k(\bar{\mathbf{w}}^k) \leq \omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k$, where $\omega_k := \prod_{j=0}^{k-1}(1-\tau_j)$ and $\Psi_k := \Psi_0 + \sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)\Psi_j$. Using the definition (25) of G_k and the definition (17) of g_γ, we can reexpress G_k as $G_k(\bar{\mathbf{w}}^k) = f(\bar{\mathbf{x}}^k) - g_{\gamma_k}(\bar{\mathbf{y}}^k) + (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2$. This expression leads to

$$f(\bar{\mathbf{x}}^k) - g_{\gamma_k}(\bar{\mathbf{y}}^k) \leq \omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k - (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2, \tag{79}$$

which is indeed (79).

Now, we notice that under Assumption A.1, the solution set $\mathscr{Y}^\star$ of the dual problem (11) is also nonempty and bounded. Moreover, the strong duality holds, i.e., $f^\star = g^\star$. Any point $(\mathbf{x}^\star, \mathbf{y}^\star) \in \mathscr{X}^\star \times \mathscr{Y}^\star$ is a primal-dual solution to (1)-(11), and is also a saddle point of $\mathscr{L}$, i.e., $\mathscr{L}(\mathbf{x}^\star, \mathbf{y}) \leq \mathscr{L}(\mathbf{x}^\star, \mathbf{y}^\star) \leq \mathscr{L}(\mathbf{x}, \mathbf{y}^\star)$ for all $\mathbf{x} \in \mathscr{X}$ and $\mathbf{y} \in \mathbb{R}^m$. These inequalities lead to the following estimate

$$f(\mathbf{x}) - g(\mathbf{y}) \geq f(\mathbf{x}) - f^\star \geq -\|\mathbf{y}^\star\|_2\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2, \;\; \forall \mathbf{x} \in \mathscr{X},\; \mathbf{y} \in \mathbb{R}^m, \tag{80}$$

which is exactly (80). Now, we combine (18), (79) and (80) to get the following:

$$-\|\mathbf{y}^\star\|_2\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq f(\bar{\mathbf{x}}^k) - f^\star \leq f(\bar{\mathbf{x}}^k) - g(\bar{\mathbf{y}}^k) \leq S_k - (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2 \leq S_k, \tag{81}$$

where $S_k := \omega_k G_0(\bar{\mathbf{w}}^0) + \gamma_k D_{\mathscr{X}}^{\mathbf{S}} - \Psi_k$. This bound is exactly (31).

Finally, we prove (32). Let $t := \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$. It follows from (81) that $-\|\mathbf{y}^\star\|_2 t \leq S_k - (1/(2\beta_k))t^2$. This inequation of t leads to $t \leq \beta_k\left[\|\mathbf{y}^\star\|_2 + \sqrt{\|\mathbf{y}^\star\|_2^2 + 2\beta_k^{-1}S_k}\right]$ provided $\beta_k\|\mathbf{y}^\star\|_2^2 + 2S_k \geq 0$. This estimate is indeed (32). □

A.2 Convergence analysis: The proof of Theorem 4.1.

Our proof of Theorem 4.1 takes the following outline:

1. We prove two key lemmas: Lemma 4.1 and Lemma 4.2. These lemmas provide conditions to update the step-size τ_k.
2. We show how to find starting points for Algorithm 1 using Lemma 4.3.
3. We provide an update rule for the step-size parameter τ_k in Lemma 4.4 based on the conditions of Lemmas 4.1 and 4.2.
4. We combine the above results to finalize the proof of Theorem 4.1.

A.2.1 The proof of Lemma 4.1: The condition for selecting step-size τ_k in (2P1D).

Let us denote by $d_k(\mathbf{w}) := \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (\beta_k/2)\|\mathbf{y}\|_2^2$ and $\bar{\mathbf{x}}_k^\star := \mathbf{x}_{\gamma_k}^\star(\bar{\mathbf{y}}^k)$. If we define

$$H_k(\mathbf{w}) := f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) + F(\bar{\mathbf{w}}^k)^T(\bar{\mathbf{w}}^k - \mathbf{w}) - d_k(\mathbf{w}), \tag{82}$$

the objective function in (25), then by the definition of G_{k+1} and $\mathscr{W} := \mathscr{X} \times \mathbb{R}^m$, we have

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) := \max_{\mathbf{w}\in\mathscr{W}} H_{k+1}(\mathbf{w}). \tag{83}$$

The proof is divided in the following steps:
Step 1: Splitting H_k and H_{k+1}. Using the definition of F in (15), we can write $F(\bar{\mathbf{w}}^k)^T(\bar{\mathbf{w}}^k - \mathbf{w}) = (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^k$. Plugging this expression into (82) we obtain

$$H_k(\mathbf{w}) := f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^k - d_k(\mathbf{w}). \tag{84}$$

Similarly to (84), we also have $H_{k+1}(\mathbf{w}) = f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^{k+1}$. Using this expression and $\bar{\mathbf{y}}^{k+1} = (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\hat{\mathbf{y}}^k$ in (2P1D) we get

$$H_{k+1}(\mathbf{w}) = f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (1-\tau_k)(\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \tau_k(\mathbf{A}\mathbf{x}-\mathbf{b})^T\hat{\mathbf{y}}^k - d_{k+1}(\mathbf{w}).$$

By adding and then subtracting $(1-\tau_k)[f(\bar{\mathbf{x}}^k) - f(\mathbf{x})]$ into this inequality, we obtain

$$\begin{aligned} H_{k+1}(\mathbf{w}) = (1-\tau_k)\big[f(\bar{\mathbf{x}}^k - f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k\big] + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - \tau_k(\mathbf{A}\mathbf{x}-\mathbf{b})^T\hat{\mathbf{y}}^k \\ + f(\bar{\mathbf{x}}^{k+1}) - (1-\tau_k)f(\bar{\mathbf{x}}^k) - \tau_k f(\mathbf{x}) - d_{k+1}(\mathbf{w}). \end{aligned} \tag{85}$$

Step 2: Estimating a lower bound for G_k. By using the definition (17) of g_γ, we have

$$f(\mathbf{x}) + (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k + \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) \geq g_{\gamma_k}(\bar{\mathbf{y}}^k) + \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_{\gamma_k}^\star(\bar{\mathbf{y}}^k)).$$

Using this inequality and $\max_{\mathbf{y}\in\mathbb{R}^m}\Big\{(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} - (\beta/2)\|\mathbf{y}\|_2^2\Big\} = (1/(2\beta))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2$, we can show that

$$\begin{aligned} G_k(\bar{\mathbf{w}}^k) &:= \max_{\mathbf{w}\in\mathscr{W}}\Big\{f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} - (\beta_k/2)\|\mathbf{y}\|_2^2\Big\} \\ &\geq f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2 + \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\bar{\mathbf{x}}_k^\star). \end{aligned} \tag{86}$$

From the second line $\hat{\mathbf{y}}^k := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b})$ of (2P1D), we also have the following equality

$$\begin{aligned} \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2 &= \|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 + 2(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b})^T\mathbf{A}(\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k) + \|\mathbf{A}(\hat{\mathbf{x}}^k - \bar{\mathbf{x}}^k)\|_2^2 \\ &= \|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 + 2\beta_{k+1}(\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k) + \|\mathbf{A}(\hat{\mathbf{x}}^k - \bar{\mathbf{x}}^k)\|_2^2. \end{aligned} \tag{87}$$

Since $\beta_{k+1} = (1-\tau_k)\beta_k$ due to (36), substituting (87) into (86) we obtain

$$\begin{aligned} f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) \leq G_k(\bar{\mathbf{w}}^k) - (1/(2\beta_{k+1}))\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 - (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k) \\ - \gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\bar{\mathbf{x}}_k^\star) - (1/(2\beta_{k+1}))\big[\|\mathbf{A}(\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k)\|_2^2 - \tau_k\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2^2\big]. \end{aligned} \tag{88}$$

Step 3: Estimating an upper bound for H_{k+1}. First, from the update rule (36), we have $\gamma_{k+1} = (1 - c_k\tau_k)\gamma_k \geq (1-\tau_k)\gamma_k$ for any $c_k \leq 1$ and $\beta_{k+1} = (1-\tau_k)\beta_k$. Hence, we can show that

$$d_{k+1}(\mathbf{w}) = \gamma_{k+1}d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 \geq (1-\tau_k)\gamma_k d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c) + (\beta_{k+1}/2)\|\mathbf{y}\|_2^2. \tag{89}$$

Second, by using $\hat{\mathbf{y}}^k := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b})$ in (2P1D), we also have the following equality

$$(\mathbf{A}\mathbf{x}-\mathbf{b})^T\hat{\mathbf{y}}^k = (\hat{\mathbf{y}}^k)^T\mathbf{A}(\hat{\mathbf{x}}^k-\mathbf{b}) + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}}^k) = (1/\beta_{k+1})\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}}^k). \tag{90}$$

Third, substituting (89), (88) and (90) into (85), we can upperbound the estimate H_{k+1} as

$$\begin{aligned}
H_{k+1}(\mathbf{w}) &\overset{(89)}{\leq} (1-\tau_k)\big[f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{S}\mathbf{x}_c)\big] + (\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 \\
&\quad - \tau_k(\mathbf{A}\mathbf{x}-\mathbf{b})^T\hat{\mathbf{y}}^k + f(\bar{\mathbf{x}}^{k+1}) - (1-\tau_k)f(\bar{\mathbf{x}}^k) - \tau_k f(\mathbf{x}) \\
&\overset{(88)+(90)}{\leq} (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + (\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 + \big[f(\bar{\mathbf{x}}^{k+1}) - (1-\tau_k)f(\bar{\mathbf{x}}^k) - \tau_k f(\mathbf{x})\big] \\
&\quad - \frac{(1-\tau_k)}{2\beta_{k+1}}\big[\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 + \|\mathbf{A}(\bar{\mathbf{x}}^k-\hat{\mathbf{x}}^k)\|_2^2 - \tau_k\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|_2^2\big] - (1-\tau_k)(\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^k-\hat{\mathbf{x}}^k) \\
&\quad - \tau_k\big[(1/\beta_{k+1})\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 - (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}}^k)\big] - (1-\tau_k)\gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{S}\bar{\mathbf{x}}_k^\star) \\
&= (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + \big[f(\bar{\mathbf{x}}^{k+1}) - (1-\tau_k)f(\bar{\mathbf{x}}^k) - \tau_k f(\mathbf{x})\big] + (\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 \\
&\quad + (\hat{\mathbf{y}}^k)^T\mathbf{A}\big[(1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\mathbf{x} - \hat{\mathbf{x}}^k\big] - (1/(2\beta_{k+1}))\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 - (1-\tau_k)\gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{S}\bar{\mathbf{x}}_k^\star) \\
&\quad - (1/(2\beta_{k+1}))\Big[(1-\tau_k)\|\mathbf{A}(\bar{\mathbf{x}}^k-\hat{\mathbf{x}}^k)\|_2^2 - (1-\tau_k)\tau_k\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|_2^2 + \tau_k\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2\Big].
\end{aligned} \tag{91}$$

Step 4: Refining the upper bound of H_{k+1}. Let $\mathbf{u} := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\mathbf{x} \in \mathscr{X}$ and

$$\mathscr{T}_{[3]} := (1/(2\beta_{k+1}))\big[(1-\tau_k)\|\mathbf{A}(\bar{\mathbf{x}}^k-\hat{\mathbf{x}}^k)\|_2^2 - (1-\tau_k)\tau_k\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|_2^2 + \tau_k\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2\big]. \tag{92}$$

First, by the convexity of f we have $f(\mathbf{u}) \leq (1-\tau_k)f(\bar{\mathbf{x}}) + \tau_k f(\mathbf{x})$. Second, from the first line of (2P1D) we have $\mathbf{u}-\hat{\mathbf{x}}^k = \tau_k(\mathbf{x}-\bar{\mathbf{x}}_k^\star)$. Third, by the strong convexity of d_b and the condition (37), we can estimate

$$(1-\tau_k)\gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{S}\bar{\mathbf{x}}_k^\star) \geq \bar{L}^g\beta_{k+1}^{-1}\tau_k^2 d_b(\mathbf{S}\mathbf{u},\mathbf{S}\bar{\mathbf{x}}_k^\star) \geq (\bar{L}^g/2)\beta_{k+1}^{-1}\tau_k^2\|\mathbf{S}(\mathbf{x}-\bar{\mathbf{x}}_k^\star)\|_2^2 \geq (\bar{L}^g/2)\beta_{k+1}^{-1}\|\mathbf{S}(\mathbf{u}-\hat{\mathbf{x}}^k)\|_2^2.$$

Finally, substituting these expressions into (91) we obtain

$$\begin{aligned}
H_{k+1}(\mathbf{w}) &\leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{u}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 \\
&\quad - (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{u}-\hat{\mathbf{x}}^k) - (\bar{L}^g/2)\beta_{k+1}^{-1}\|\mathbf{S}(\mathbf{u}-\hat{\mathbf{x}}^k)\|_2^2 - (1/(2\beta_{k+1}))\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 - \mathscr{T}_{[3]}.
\end{aligned} \tag{93}$$

Step 5: Final touches on the upper bound of H_{k+1}. By the third line of (2P1D), we have $\bar{\mathbf{x}}^{k+1} := \mathrm{prox}_{\mathbf{S}f}(\hat{\mathbf{x}}^k, \hat{\mathbf{y}}^k; \beta_{k+1})$. If we define $\mathscr{H}_{\beta_{k+1}}(\mathbf{u}) := f(\mathbf{u}) + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{u}-\hat{\mathbf{x}}^k) + (\bar{L}^g/(2\beta_{k+1}))\|\mathbf{S}(\mathbf{u}-\hat{\mathbf{x}}^k)\|_2^2$, then, by (34), we have

$$\mathscr{H}_{\beta_{k+1}}(\mathbf{u}) \geq \mathscr{H}_{\beta_{k+1}}(\bar{\mathbf{x}}^{k+1}), \ \ \forall \mathbf{u} \in \mathscr{X}. \tag{94}$$

On the other hand, since $\max\limits_{\mathbf{y}\in\mathbb{R}^m}\big\{(\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2\big\} = (1/(2\beta_{k+1}))\|\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b}\|_2^2$, one has

$$(\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 \leq (1/(2\beta_{k+1}))\|\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b}\|_2^2, \ \ \forall \mathbf{y} \in \mathbb{R}^m. \tag{95}$$

Substituting (95) and (94) into (93) we get

$$\begin{aligned}
H_{k+1}(\mathbf{w}) &\leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + f(\bar{\mathbf{x}}^{k+1}) - f(\bar{\mathbf{x}}^{k+1}) - (1/(2\beta_{k+1}))\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 \\
&\quad - (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^k) - (\bar{L}^g/(2\beta_{k+1}))\|\mathbf{S}(\bar{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^k)\|_2^2 + (1/(2\beta_{k+1}))\|\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b}\|_2^2 - \mathscr{T}_{[3]}.
\end{aligned} \tag{96}$$

By the condition (35) with $\hat{\mathbf{x}} = \hat{\mathbf{x}}^k$, $\mathbf{x} = \bar{\mathbf{x}}^{k+1}$ and $\hat{\mathbf{y}}^k := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b})$, we have

$$(2\beta_{k+1})^{-1}\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2 + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^k) + (2\beta_{k+1})^{-1}\bar{L}^g\|\mathbf{S}(\bar{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^k)\|_2^2 \geq (2\beta_{k+1})^{-1}\|\mathbf{A}\bar{\mathbf{x}}^{k+1}-\mathbf{b}\|_2^2. \tag{97}$$

Substituting this inequality into (96) we finally get

$$H_{k+1}(\mathbf{w}) \le (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - \mathscr{T}_{[3]}. \tag{98}$$

Step 6: We simplify $\mathscr{T}_{[3]}$ and prove (27). From the definition (92) of $\mathscr{T}_{[3]}$, we can estimate

$$\begin{aligned}\mathscr{T}_{[3]} &:= (2\beta_{k+1})^{-1}\left[(1-\tau_k)\|\mathbf{A}(\bar{\mathbf{x}}^k-\hat{\mathbf{x}}^k)\|_2^2 - \tau_k(1-\tau_k)\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|_2^2 + \tau_k\|\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}\|_2^2\right]\\ &= (2\beta_{k+1})^{-1}\|(\mathbf{A}\hat{\mathbf{x}}^k-\mathbf{b}) - (1-\tau_k)(\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})\|_2^2\\ &\overset{\text{(2P1D)(line 1)}}{=} (2\beta_{k+1})^{-1}\tau_k^2\|\mathbf{A}\bar{\mathbf{x}}_k^\star-\mathbf{b}\|_2^2.\end{aligned} \tag{99}$$

Substituting (99) into (98) and taking the maximization over $\mathscr{W}$ we obtain

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) = \max_{\mathbf{w}\in\mathscr{W}} H_{k+1}(\mathbf{w}) \le (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - (\tau_k^2/(2\beta_{k+1}))\|\mathbf{A}\bar{\mathbf{x}}_k^\star-\mathbf{b}\|_2^2,$$

which is indeed (27). □

A.2.2 The proof of Lemma 4.2: The condition for selecting step-size τ_k in (1P2D).

Let us denote by $\bar{\mathbf{y}}_k^\star := \mathbf{y}_{\beta_k}^\star(\bar{\mathbf{x}}^k)$, $\bar{\mathbf{x}}_k^\star := \mathbf{x}_{\gamma_{k+1}}^\star(\bar{\mathbf{y}}^k)$ and $\hat{\mathbf{x}}_k^\star := \mathbf{x}_{\gamma_{k+1}}^\star(\hat{\mathbf{y}}^k)$.
Step 1: Estimate G_k. By using H_k as in the proof of Lemma 4.1[(85)], we have

$$\begin{aligned}G_k(\bar{\mathbf{w}}^k) &= \max_{\mathbf{w}\in\mathscr{W}}\left\{f(\bar{\mathbf{x}}^k) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - d_k(\mathbf{w})\right\}\\ &\ge f(\bar{\mathbf{x}}^k) + \max_{\mathbf{x}\in\mathscr{X}}\left\{-f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{x}_c)\right\} + \max_{\mathbf{y}\in\mathbb{R}^m}\left\{(\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})^T\mathbf{y} - \frac{\beta_k}{2}\|\mathbf{y}\|_2^2\right\}.\end{aligned} \tag{100}$$

Now, since $\mathbf{s}^T\mathbf{y} - (\beta/2)\|\mathbf{y}\|_2^2 = (1/(2\beta))\|\mathbf{s}\|_2^2 - (\beta/2)\|\mathbf{y}-(1/\beta)\mathbf{s}\|_2^2$ for all $\mathbf{y},\mathbf{s}\in\mathbb{R}^m$, we have

$$(\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})^T\mathbf{y} - (\beta_k/2)\|\mathbf{y}\|_2^2 + (\beta/2)\|\mathbf{y}-\bar{\mathbf{y}}_k^\star\|_2^2 \le \max_{\mathbf{y}\in\mathbb{R}^m}\left\{(\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})^T\mathbf{y} - (\beta_k/2)\|\mathbf{y}\|_2^2\right\}.$$

Substituting this estimate into (100) we get

$$\begin{aligned}G_k(\bar{\mathbf{w}}^k) \ge f(\bar{\mathbf{x}}^k) &+ \max_{\mathbf{x}\in\mathscr{X}}\left\{-f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma_k d_b(\mathbf{S}\mathbf{x},\mathbf{x}_c)\right\}\\ &+ (\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b})^T\mathbf{y} - (\beta_k/2)\|\mathbf{y}\|_2^2 + (\beta/2)\|\mathbf{y}-\bar{\mathbf{y}}_k^\star\|_2^2.\end{aligned} \tag{101}$$

Step 2: Properties of g_γ. Let $\varphi_\gamma(\mathbf{y}) := \max_{\mathbf{x}\in\mathscr{X}}\left\{-f(\mathbf{x}) - (\mathbf{A}\mathbf{x}-\mathbf{b})^T\bar{\mathbf{y}}^k - \gamma d_b(\mathbf{S}\mathbf{x},\mathbf{S}\mathbf{x}_c)\right\}$. It is clear that $\varphi_{\gamma_k}(\mathbf{y}) \equiv -g_{\gamma_k}(\mathbf{y})$, which is convex and smooth. Hence, by Definition 3.1, we have

$$\left\{\begin{array}{ll}\varphi_\gamma(\mathbf{y}) & \ge \varphi_\gamma(\hat{\mathbf{y}}) + \nabla\varphi_\gamma(\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}}),\\ \varphi_\gamma(\mathbf{y}) & \le \varphi_\gamma(\hat{\mathbf{y}}) + \nabla\varphi_\gamma(\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}}) + (L_\gamma^g/2)\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2,\ \ \forall\mathbf{y},\hat{\mathbf{y}}\in\mathbb{R}^m,\\ \varphi_{\bar\gamma}(\mathbf{y}) & \ge \varphi_\gamma(\mathbf{y}) + (\gamma-\bar\gamma)d_b(\mathbf{S}\mathbf{x}_\gamma^\star(\mathbf{y}),\mathbf{S}\mathbf{x}_c),\ \ \forall\,\bar\gamma,\gamma>0.\end{array}\right. \tag{102}$$

Here the first inequality follows from the convexity of φ_γ, while the second follows from the Lipschitz continuity of $\nabla\varphi_\gamma$. We prove the third inequality. The function $s(\mathbf{x},\gamma) := -f(\mathbf{x}) - \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) - \gamma d_b(\mathbf{S}\mathbf{x},\mathbf{S}\mathbf{x}_c)$ is concave with respect to $\mathbf{x}$ and linear with respect to γ. It is clear that $\varphi_\gamma(\mathbf{y}) = \max_{\mathbf{x}\in\mathscr{X}}\{s(\mathbf{x},\gamma)\}$, which is convex with respect to γ [13]. Moreover, its derivative with respect to γ is given by $-d_b(\mathbf{S}\mathbf{x}_\gamma^\star(\mathbf{y}),\mathbf{S}\mathbf{x}_c) \ge 0$. This function is nonincreasing, which leads to the first inequality of (102).

Step 3: A refinement of G_k. By the definition of $\hat{\mathbf{x}}_k^\star$ and $\nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_k^\star$, we can express

$$\begin{aligned}-f(\hat{\mathbf{x}}_k^\star) - (\mathbf{A}\hat{\mathbf{x}}_k^\star - \mathbf{b})^T\mathbf{y} &= -f(\hat{\mathbf{x}}_k^\star) - (\mathbf{A}\hat{\mathbf{x}}_k^\star - \mathbf{b})^T\hat{\mathbf{y}}^k - (\mathbf{A}\hat{\mathbf{x}}_k^\star - \mathbf{b})^T(\mathbf{y} - \hat{\mathbf{y}}^k)\\ &= \varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k)^T(\mathbf{y} - \hat{\mathbf{y}}^k) + \gamma_{k+1}d_b(\mathbf{S}\hat{\mathbf{x}}_k^\star, \mathbf{S}\mathbf{x}_c). \end{aligned} \tag{103}$$

Multiplying (101) by $1-\tau_k$ and then using the definition of φ_γ, we have

$$(1-\tau_k)G_k(\bar{\mathbf{w}}^k) = (1-\tau_k)\Big[\varphi_{\gamma_k}(\bar{\mathbf{y}}^k) + f(\bar{\mathbf{x}}^k) + (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} + \frac{\beta_k}{2}\|\mathbf{y} - \bar{\mathbf{y}}_k^\star\|_2^2 - \frac{\beta_k}{2}\|\mathbf{y}\|_2^2\Big]. \tag{104}$$

Using the third inequality of (103) with $\bar{\gamma} = \gamma_k$ and $\gamma = \gamma_{k+1} = (1 - c_k\tau_k)\gamma_k$ and (103) into (104) we obtain

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq (1-\tau_k)\big[\varphi_{\gamma_{k+1}}(\bar{\mathbf{y}}^k) + f(\bar{\mathbf{x}}^k) + (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y} + \frac{\beta_k}{2}\|\mathbf{y} - \bar{\mathbf{y}}_k^\star\|_2^2 - \frac{\beta_k}{2}\|\mathbf{y}\|_2^2 - \tau_k c_k\gamma_k d_b(\mathbf{S}\bar{\mathbf{x}}_k^\star, \mathbf{S}\mathbf{x}_c)\big]\\ + \tau_k\Big[f(\hat{\mathbf{x}}_k^\star) + (\mathbf{A}\hat{\mathbf{x}}_k^\star - \mathbf{b})^T\mathbf{y} + \varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k)^T(\mathbf{y} - \hat{\mathbf{y}}^k) + \gamma_{k+1}d_b(\mathbf{S}\hat{\mathbf{x}}_k^\star, \mathbf{S}\mathbf{x}_c)\Big]. \end{aligned} \tag{105}$$

Now, using the first line of (103), we have $\varphi_{\gamma_{k+1}}(\bar{\mathbf{y}}^k) \geq \varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k)$. On the other hand, by the convexity of f and the second line of (1P2D), we easily get $f(\bar{\mathbf{x}}^{k+1}) = f((1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\hat{\mathbf{x}}_k^\star) \leq (1 - \tau_k)f(\bar{\mathbf{x}}^k) + \tau_k f(\hat{\mathbf{x}}_k^\star)$. Using these inequalities and $\bar{\mathbf{x}}^{k+1} = (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\hat{\mathbf{x}}_k^\star$ into (105) we can further estimate

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) &\geq \varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k)^T\big[(1-\tau_k)(\bar{\mathbf{y}}^k + \tau_k\mathbf{y} - \hat{\mathbf{y}}^k\big] + f(\bar{\mathbf{x}}^{k+1}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y}\\ &\quad + \frac{(1-\tau_k)\beta_k}{2}\big[\|\mathbf{y} - \bar{\mathbf{y}}_k^\star\|_2^2 - \|\mathbf{y}\|_2^2\big] + \tau_k\gamma_{k+1}d_b(\mathbf{S}\hat{\mathbf{x}}_\star^k, \mathbf{S}\mathbf{x}_c) - (1-\tau_k)\tau_k c_k\gamma_k d_b(\mathbf{S}\bar{\mathbf{x}}_k^\star, \mathbf{S}\mathbf{x}_c). \end{aligned} \tag{106}$$

Step 4: We prove (27). Let $\mathbf{v} := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\mathbf{y} \in \mathbb{R}^m$. Using the first line of (1P2D), we can show that $\mathbf{v} - \hat{\mathbf{y}}^k := \tau_k(\mathbf{y} - \bar{\mathbf{y}}_k^\star)$. Substituting $\mathbf{v}$ into (106) and taking the maximization over $\mathbb{R}^m$, we get

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq \max_{\mathbf{v}\in\mathbb{R}^m}\Big\{&\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_{k+1}}(\hat{\mathbf{y}}^k)^T(\mathbf{v} - \hat{\mathbf{y}}^k) + (\beta_{k+1}/\tau_k^2)\|\mathbf{v} - \hat{\mathbf{y}}^k\|_2^2\Big\}\\ &+ f(\bar{\mathbf{x}}^{k+1}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 + \mathscr{T}_{[4]}, \end{aligned} \tag{107}$$

where $\mathscr{T}_{[4]} := \tau_k(1-\tau_k)\gamma_k\big[d_b(\mathbf{S}\hat{\mathbf{x}}_\star^k, \mathbf{S}\mathbf{x}_c) - c_k d_b(\mathbf{S}\bar{\mathbf{x}}_k^\star, \mathbf{S}\mathbf{x}_c)\big]$.

From the condition $\beta_{k+1}\gamma_{k+1} \geq \bar{L}^g\tau_k^2$ in (38), we have $\beta_{k+1}\tau_k^{-2} \geq \bar{L}^g\gamma_{k+1}^{-1} = L^g_{\gamma_{k+1}}$. Using this inequality, the second line of (1P2D) and the second inequality of (102) with $\gamma = \gamma_{k+1}$, $\hat{\mathbf{y}} = \hat{\mathbf{y}}^k$ and $\mathbf{y} = \bar{\mathbf{y}}^{k+1}$, we can further refine (107) as

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) &\geq \varphi_{\gamma_{k+1}}(\bar{\mathbf{y}}^{k+1}) + f(\bar{\mathbf{x}}^{k+1}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 + \mathscr{T}_{[4]}\\ &\geq f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^{k+1} - d_{k+1}(\mathbf{w}) + \mathscr{T}_{[4]}\\ &= H_{k+1}(\mathbf{w}) + \mathscr{T}_{[4]}. \end{aligned} \tag{108}$$

Since the left-hand side of (108) is constant, by maximizing over $\mathbf{w} \in \mathscr{W}$ the right-hand side of this inequality, we finally get $G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - \mathscr{T}_{[4]}$, which is indeed (27). □

A.2.3 The proof of Lemma 4.3: Finding starting points for Algorithm 1.

From the definition of H_k in the proof of Lemma 4.1[(85)] and the definition of g_γ, we can show that

$$\begin{aligned}G_0(\bar{\mathbf{w}}^0) &= \max_{\mathbf{w}\in\mathscr{W}}\big\{f(\bar{\mathbf{x}}^0) - f(\mathbf{x}) - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^0 + (\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b})^T\mathbf{y} - d_0(\mathbf{w})\big\}\\ &= \max_{\mathbf{y}\in\mathbb{R}^m}\Big\{f(\bar{\mathbf{x}}^0) + (\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b})^T\mathbf{y} - \frac{\beta_0}{2}\|\mathbf{y}\|_2^2\Big\} - \min_{\mathbf{x}\in\mathscr{X}}\Big\{f(\mathbf{x}) + (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^0 + \gamma_0 d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}_c)\Big\}\\ &= \max_{\mathbf{y}\in\mathbb{R}^m}\Big\{f(\bar{\mathbf{x}}^0) + (\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b})^T\mathbf{y} - \frac{\beta_0}{2}\|\mathbf{y}\|_2^2 - g_{\gamma_0}(\bar{\mathbf{y}}^0)\Big\}. \end{aligned} \tag{109}$$

By the definition of $\bar{\mathbf{x}}^0$ and $\mathbf{y}^c := 0^m$, we have $g_{\gamma_0}(\mathbf{y}^c) = f(\bar{\mathbf{x}}^0) + \gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c)$. By Definition 3.1, $\nabla g_\gamma(\cdot)$ is $\bar{L}^g/\gamma_0$-Lipschitz continuous, by [48, Theorem 2.1.5], we have $g_{\gamma_0}(\bar{\mathbf{y}}^0) \geq g_{\gamma_0}(\mathbf{y}^c) + \nabla g_{\gamma_0}(\mathbf{y}^c)^T(\bar{\mathbf{y}}^0 - \mathbf{y}_c) - \frac{\bar{L}^g}{2\gamma_0}\|\bar{\mathbf{y}}^0 - \mathbf{y}_c\|_2^2$. Moreover, $\nabla g_{\gamma_0}(\mathbf{y}^c) = \mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}$ for $\bar{\mathbf{y}}^0 = (1/\beta_0)\left(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\right)$. Hence, $g_{\gamma_0}(\bar{\mathbf{y}}^0) \geq f(\bar{\mathbf{x}}^0) + (\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b})^T\bar{\mathbf{y}}^0 - \frac{\bar{L}^g}{2\gamma_0\beta_0^2}\|\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\|_2^2$. Using this inequality into (109), we can further estimate

$$\begin{aligned} G_0(\bar{\mathbf{w}}^0) &\leq \max_{\mathbf{y}\in\mathbb{R}^m}\Big\{(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b})^T\mathbf{y} - \frac{\beta_0}{2}\|\mathbf{y}\|_2^2 + \frac{\bar{L}^g}{2\gamma_0\beta_0^2}\|\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\|_2^2 - \gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c)\Big\} \\ &\leq -\frac{1}{2\beta_0}\Big(2 - \frac{\bar{L}^g}{\beta_0\gamma_0}\Big)\|\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}\|_2^2 - \gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c), \end{aligned}$$

which leads to $G_0(\bar{\mathbf{w}}^0) \leq -\gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}^0, \mathbf{S}\mathbf{x}_c)$ provided that $\beta_0\gamma_0 \geq \bar{L}^g$. The statement (40) of Lemma 4.3 can be proved similarly. □

A.2.4 The proof of Lemma 4.4: Update rule for step-size parameter τ_k.

For any $c_{k+1} \leq 1$ and $a_k \geq 0$, we have $1 + c_{k+1} + 2a_k \leq 1 + c_{k+1} + \sqrt{4a_k^2 + (1-c_{k+1})^2} \leq 2a_k + 1 - c_{k+1} + 1 + c_{k+1} = 2a_k + 2$. From (41), we can easily show that $a_k + (c_{k+1}+1)/2 \leq a_{k+1} \leq a_k + 1$. By induction, we can derive from this estimate that

$$a_0 + k/2 + (1/2)\sum_{i=1}^{k} c_i \leq a_k \leq a_0 + k.$$

On the other hand, from (41) we have $a_0 := \left(\sqrt{(1+c_0)^2 + 4(1-c_0)} + 1 + c_0\right)/2$. Combining two last expressions and $s_k := \sum_{i=1}^k c_i$, we obtain (43). The estimate (44) follows from the relation $\beta_{k+1}\gamma_{k+1} = \bar{L}^g\tau_k^2 = \bar{L}^g a_k^{-2}$ and (43).

Now, let us consider the case $c_k = 0$ for all $k \geq 0$. Then, the update rule for γ_k becomes $\gamma_{k+1} := \gamma_k = \gamma_0 = \bar{L}^g/\beta_0$ for all $k \geq 0$. Moreover, we have $a_0 = (1+\sqrt{5})/2$. Then the first line of (45) follows directly from (44) and $1 < a_0 < 2$.

If $c_k = 1$ for all $k \geq 0$ then $a_0 = 2$ and $\tau_0 = 0.5$. Moreover, we have $(1-\tau_{k+1})^2\tau_k^2 = \tau_{k+1}^2$, which leads to $(1-\tau_{k+1}) = \tau_{k+1}/\tau_k$. Therefore, $\beta_{k+1} = \beta_0\prod_{i=0}^k(1-\tau_i) = \beta_0(1-\tau_0)\prod_{i=1}^k\frac{\tau_i}{\tau_{i-1}} = \beta_0(1-\tau_0)\frac{\tau_k}{\tau_0} = \beta_0 a_k^{-1}$. Moreover, from (43) we have $k+2 = k+a_0 \leq a_k \leq k+a_0 = k+2$. Combining the last inequality and this equality we obtain the second line of (45). □

A.2.5 The full-proof of Theorem 4.1.

Under Assumption A.1, by the well-known properties of augmented Lagrangian function $\mathscr{L}_\gamma$, see, e.g. [10], we have

$$\mathscr{L}_\gamma(\mathbf{x}^\star, \mathbf{y}) \leq \mathscr{L}_\gamma(\mathbf{x}^\star, \mathbf{y}^\star) \equiv \mathscr{L}(\mathbf{x}^\star, \mathbf{y}^\star) = f^\star = g^\star \leq \mathscr{L}_\gamma(\mathbf{x}, \mathbf{y}^\star)$$

for all $\mathbf{x} \in \mathscr{X}, \mathbf{y} \in \mathbf{R}^m$, $(\mathbf{x}^\star, \mathbf{y}^\star) \in \mathscr{W}^\star$ and $\gamma > 0$. This expression leads to

$$\tilde{g}_\gamma(\mathbf{y}) \leq f(\mathbf{x}) + (\mathbf{A}\mathbf{x} - \mathbf{b})^T\mathbf{y}^\star + (\gamma/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \leq f(\mathbf{x}) + \|\mathbf{y}^\star\|_2\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 + (\gamma/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Hence, for any $\mathbf{y}^\star \in \mathscr{Y}^\star$, we obtain

$$f(\mathbf{x}) - \tilde{g}_\gamma(\mathbf{y}) \geq f(\mathbf{x}) - f^\star \geq -\|\mathbf{y}^\star\|_2\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 - (\gamma/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \;\; \forall \mathbf{x} \in \mathscr{X}, \mathbf{y} \in \mathrm{dom}\left(g_\gamma\right). \tag{110}$$

Let $t := \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2$. By combining (110) and (79) we obtain $\frac{(1-\gamma_k\beta_k)}{\beta_k}t^2 - 2\|\mathbf{y}^\star\|_2 t - 2(\omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k) \leq 0$. Since $\gamma_k\beta_k \leq \bar{L}^g\tau_{k-1}^2 < \bar{L}^g \equiv 1$, we can show that

$$\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \Big(\frac{\beta_k}{1-\beta_k\gamma_k}\Big)\Big[\|\mathbf{y}^\star\|_2 + \Big(\|\mathbf{y}^\star\|_2^2 + \frac{2(\omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k)(1-\beta_k\gamma_k)}{\beta_k}\Big)^{1/2}\Big]. \tag{111}$$

To prove (46), we note that by setting $c_k := 0$ for all $k \geq 0$ in Lemma 4.4, we can derive $\frac{\beta_k}{1-\gamma_k\beta_k} \leq \frac{4\sqrt{\bar{L}^g}}{k^2-4} \leq \frac{4}{(k+1)^2}$ for $k \geq 0$. In addition, $\omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k \leq 0$ due to Lemma 4.3. Using these estimates into (111), we obtain $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{8D^\star_{\mathscr{Y}}}{(k+1)^2}$, which is the first inequality of (46).

From (79) and (110) we have $f(\bar{\mathbf{x}}^k) - f^\star \leq f(\mathbf{x}) - \tilde{g}_{\gamma_k}(\bar{\mathbf{y}}^k) \leq 0$. This inequality and (110) implies the second inequality of (46).

Next, we prove (47). By Lemma 3.3 we have $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \beta_k\|\mathbf{y}^\star\| + \sqrt{\beta_k^2\|\mathbf{y}^\star\|^2 + 2\beta_k\gamma_k D_k} \leq 2\beta_k D^\star_{\mathscr{Y}} + \sqrt{2\gamma_k\beta_k D^{\mathbb{I}}_{\mathscr{X}}}$, where $\mathbf{y}^\star$ is one minimum norm element of $\mathscr{Y}^\star$. By Lemma 4.4, we have $\beta_k\gamma_k = \frac{\bar{L}^g}{(k+1)^2}$ and $\beta_k = \frac{\sqrt{\bar{L}^g}}{k+1}$. Combines these equalities, we obtain the first inequality in (47). The second inequality of (47) follows from Lemma 3.3 and $\beta_k = \frac{\sqrt{\bar{L}^g}}{k+1}$.

To prove (48), we first see from Lemma 4.3 and Theorem 4.2 that the sequence $\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\}$ generated by Algorithm 1 maintains the condition (27). By Lemma 3.3 and Lemma 4.4 we have

$$\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{8\bar{L}_g}{\gamma_0(k+1)^2}\|\mathbf{y}^\star\| + \frac{\sqrt{8\bar{L}_g D_k}}{(k+1)}.$$

By the definition of $D_{\mathscr{Y}^\star}$ and the choice of γ_0, we obtain from this inequality the first estimate of (48). The second estimate of (48) immediately follows from (33) and the choice of γ_0. □

A.3 The proof of Corollary 5.1: Strong convexity case.

For simplicity of presentation, we divide this proof into few steps.
Step 1: The proof of Corollary 5.1 for the (1P2D_s) *scheme.* The proof of the two first estimates in Corollary 5.1 for (1P2D_s) can be done similarly to [67, Theorem 4], where we can show that $-\frac{4L^g_f}{(k+2)^2}(D^\star_{\mathscr{Y}})^2 \leq f(\bar{\mathbf{x}}^k) - g(\bar{\mathbf{y}}^k) \leq 0$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{4L^g_f}{(k+2)^2}D^\star_{\mathscr{Y}}$. However, we have $-\|\mathbf{y}^\star\|\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq f(\mathbf{x}) - f^\star \leq f(\mathbf{x}) - g(\mathbf{y})$ for $\mathbf{x} \in \mathscr{X}$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{y}^\star \in \mathscr{Y}^\star$ due to (80). The first inequality implies the second inequality of Corollary 5.1.
Step 2: The proof of Corollary 5.1 for the (2P1D_s) *scheme.* Next, we prove the first two estimates in Corollary 5.1 for the scheme (2P1D_s). Let $\hat{\mathbf{y}}^k := \beta_k^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b})$. By applying the same argument as the proof of (93) in Lemma 4.1 to the scheme (2P1D_s), we obtain

$$\begin{aligned} H_{k+1}(\mathbf{w}) \leq{}& (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{u}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\beta_k/2)\|\mathbf{y}\|_2^2 \\ &- (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{u} - \hat{\mathbf{x}}^k) - (1/(2\beta_k))\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 - ((1-\tau_k)\sigma_f/2)\|\mathbf{x} - \mathbf{x}^*(\bar{\mathbf{y}}^k)\|_2^2 - \mathscr{T}_{[3]}, \end{aligned} \tag{112}$$

where $\mathscr{T}_{[3]}$ is defined by (92).

Let us assume that $\beta_k(1-\tau_k)\sigma_f \geq \|\mathbf{A}\|_2^2\tau_k^2$. By using this relation, $\mathbf{x} - \mathbf{x}^\star(\bar{\mathbf{y}}^k) = \tau_k^{-1}(\mathbf{u} - \hat{\mathbf{x}}^k)$ and (95), we can further modify (112) as

$$\begin{aligned} H_{k+1}(\mathbf{w}) \leq{}& (1-\tau_k)G_k(\bar{\mathbf{w}}^k) + f(\bar{\mathbf{x}}^{k+1}) - \big[f(\mathbf{u}) + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{u} - \hat{\mathbf{x}}^k) + (\|\mathbf{A}\|_2^2/(2\beta_k))\|\mathbf{u} - \hat{\mathbf{x}}^k\|_2^2\big] \\ &+ (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b}\|_2^2 - (1/(2\beta_k))\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 - \mathscr{T}_{[3]}. \end{aligned} \tag{113}$$

Using the second line $\bar{\mathbf{x}}^{k+1} = \text{prox}_{\mathbb{I}_f}(\hat{\mathbf{x}}^k, \hat{\mathbf{y}}^k; \beta_k)$ of (2P1D_s), we have

$$f(\mathbf{u}) + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\mathbf{u} - \hat{\mathbf{x}}^k) + \frac{\|\mathbf{A}\|_2^2}{2\beta_k}\|\mathbf{u} - \hat{\mathbf{x}}^k\|_2^2 \geq f(\bar{\mathbf{x}}^{k+1}) + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k) + \frac{\|\mathbf{A}\|_2^2}{2\beta_k}\|\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2.$$

Substituting this inequality into (113) we reach

$$H_{k+1}(\mathbf{w}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - (1/(2\beta_k))\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 - (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k) - (\|\mathbf{A}\|_2^2/(2\beta_k))\|\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 + (1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b}\|_2^2 - \mathscr{T}_{[3]}. \tag{114}$$

Now, we use the expression (87) for $\bar{\mathbf{x}}^k := \bar{\mathbf{x}}^{k+1}$, we can estimate

$$(1/(2\beta_k))\|\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b}\|_2^2 \leq (1/(2\beta_k))\|\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}\|_2^2 + (\hat{\mathbf{y}}^k)^T\mathbf{A}(\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k) + (\|\mathbf{A}\|_2^2/(2\beta_k))\|\bar{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2.$$

Substituting this inequality into (114), we finally obtain

$$H_{k+1}(\mathbf{w}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - \mathscr{T}_{[3]}.$$

Since $\mathscr{T}_{[3]} = (2\beta_{k+1})^{-1}\tau_k^2\|\mathbf{A}\mathbf{x}^\star(\bar{\mathbf{y}}) - \mathbf{b}\|_2^2$ due to (99), by maximizing the last inequality over $\mathbf{w} \in \mathscr{W}$, we obtain $G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - (2\beta_{k+1})^{-1}\tau_k^2\|\mathbf{A}\mathbf{x}^\star(\bar{\mathbf{y}}) - \mathbf{b}\|_2^2$. This inequality shows that the condition (27) satisfies with $\psi_k := (2\beta_{k+1})^{-1}\tau_k^2\|\mathbf{A}\mathbf{x}^\star(\bar{\mathbf{y}}) - \mathbf{b}\|_2^2 \geq 0$.

To complete the proof, we derive the condition on updating τ_k from $\frac{(1-\tau_k)\sigma_f}{\tau_k^2} \geq \frac{\|\mathbf{A}\|_2^2}{\beta_k}$. Indeed, since $\beta_{k+1} = (1-\tau_k)\beta_k$, we have $\frac{(1-\tau_{k+1})\sigma_f}{\tau_{k+1}^2} \geq \frac{\|\mathbf{A}\|_2^2}{\beta_{k+1}}$ by induction. Combining the two last conditions with equality, we obtain $(1-\tau_{k+1})\tau_k^2 = \tau_{k+1}^2$. This relation leads to $\tau_{k+1} = \tau_k\big(\sqrt{\tau_k^2+4} - \tau_k\big)/2$ as given in Corollary 5.1. Now, we use the same argument as the proof of (1P2D$_s$) to obtain the worst-case bounds in Corollary 5.1.

Step 3: The proof for the bound on $\{\bar{\mathbf{x}}^k\}$ in Corollary 5.1. Finally, we prove the last estimate of (50). Indeed, by the strong convexity of f, we have $f(\bar{\mathbf{x}}^k) - f^\star \geq \xi_f(\mathbf{x}^\star)^T(\bar{\mathbf{x}}^k - \mathbf{x}^\star) + \frac{\sigma_f}{2}\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2^2$, where $\xi_f(\mathbf{x}^\star) \in \partial f(\mathbf{x}^\star)$ is one subgradient of f at $\mathbf{x}^\star$. On the other hand, since $\mathbf{x}^\star$ is the optimal solution of (1), using the optimality condition of this problem, we have $(\xi_f(\mathbf{x}^\star) + \mathbf{A}^T\mathbf{y}^\star)^T(\mathbf{x} - \mathbf{x}^\star) \geq 0$ for any $\mathbf{x} \in \mathscr{X}$ and $\mathbf{y}^\star \in \mathscr{Y}^\star$ and $\mathbf{A}\mathbf{x}^\star = \mathbf{b}$. Using these expressions, we can show that

$$f(\bar{\mathbf{x}}^k) - f^\star \geq \frac{\sigma_f}{2}\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2^2 - (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y}^\star \geq \frac{\sigma_f}{2}\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2^2 - \|\mathbf{y}^\star\|_2\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2.$$

This estimate leads to $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\|_2^2 \leq \frac{2}{\sigma_f}[f(\bar{\mathbf{x}}^k) - f^\star] + \frac{2\|\mathbf{y}^\star\|_2}{\sigma_f}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|_2 \leq \frac{16L_f^g}{\sigma_f(k+2)^2}(D_{\mathscr{Y}}^\star)^2$, which is indeed the third estimate in Corollary 5.1. □

A.4 The proof of Lemma 5.1: Dual function g_γ is strongly convex

Let $\mathbf{x}_{p,\gamma}^\star(\mathbf{y})$ be the solution of the minimization problem in (51). Since this problem is an unconstrained convex minimization, we can write its optimality condition as

$$\mathbf{A}_p^T\mathbf{y} + \nabla f_p(\mathbf{x}_{p,\gamma}^\star(\mathbf{y})) + \gamma\nabla d_p(\mathbf{x}_{p,\gamma}^\star(\mathbf{y}), \mathbf{x}_p^c) = 0. \tag{115}$$

Moreover, we have $\nabla g_\gamma^p(\mathbf{y}) := \mathbf{A}_p\mathbf{x}_{p,\gamma}^\star(\mathbf{y})$. Since ∇f_p is L_{f_p}-Lipschitz gradient and $\nabla d_p(\cdot, \mathbf{x}_p^c)$ is 1-Lipschitz continuous, the function $\psi_p(\cdot) := \nabla f_p(\cdot) + \gamma\nabla d_p(\cdot, \mathbf{x}_p^c)$ is $(L_{f_p} + \gamma)$-Lipschitz continuous. Using Baillon-Haddad's theorem [2, Corollary 18.16], we obtain that $\psi_p(\cdot)$ is $(L_{f_p} + \gamma)^{-1}$-co-coercive, i.e.,:

$$(\psi_p(\mathbf{x}_p) - \psi_p(\hat{\mathbf{x}}_p))^T(\mathbf{x}_p - \hat{\mathbf{x}}_p) \geq (L_{f_p} + \gamma)^{-1}\|\psi_p(\mathbf{x}_p) - \psi_p(\hat{\mathbf{x}}_p)\|_2^2, \ \ \forall \mathbf{x}_p, \hat{\mathbf{x}}_p \in \mathbb{R}^{n_p}. \tag{116}$$

Now, let g_γ^p be defined by (51), we estimate the term $\mathscr{A} := (\nabla g_\gamma^p(\mathbf{y}) - \nabla g_\gamma^p(\hat{\mathbf{y}}))^T(\mathbf{y}-\hat{\mathbf{y}})$ as follows:

$$\begin{aligned}(\nabla g_\gamma^p(\mathbf{y}) - \nabla g_\gamma^p(\hat{\mathbf{y}}))^T(\mathbf{y}-\hat{\mathbf{y}}) &= (\mathbf{A}_p\mathbf{x}^\star_{p,\gamma}(\mathbf{y}) - \mathbf{A}_p\mathbf{x}^\star_{p,\gamma}(\hat{\mathbf{y}}))^T(\mathbf{y}-\hat{\mathbf{y}})\\ &= (\mathbf{y}-\hat{\mathbf{y}})^T\mathbf{A}_p(\mathbf{x}^\star_{p,\gamma}(\mathbf{y}) - \mathbf{x}^\star_{p,\gamma}(\hat{\mathbf{y}}))\\ &\overset{(115)}{=} -\left(\psi_p(\mathbf{x}^\star_{p,\gamma}(\mathbf{y})) - \psi_p(\mathbf{x}^\star_{p,\gamma}(\hat{\mathbf{y}}))\right)^T(\mathbf{x}^\star_{p,\gamma}(\mathbf{y}) - \mathbf{x}^\star_{p,\gamma}(\hat{\mathbf{y}}))\\ &\overset{(116)}{\leq} -(L_{f_p}+\gamma)^{-1}\|\psi_p(\mathbf{x}^\star_{p,\gamma}(\mathbf{y})) - \psi_p(\mathbf{x}^\star_{p,\gamma}(\hat{\mathbf{y}})\|_2^2\\ &\overset{(115)}{=} -(L_{f_p}+\gamma)^{-1}\|\mathbf{A}_p^T(\mathbf{y}-\hat{\mathbf{y}}\|_2^2\\ &\leq -(L_{f_p}+\gamma)^{-1}\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p)\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2.\end{aligned}$$

This inequality shows that g_γ^p is strongly concave with the parameter $\sigma_{g_\gamma^p} := (L_{f_p}+\gamma)^{-1}\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p) > 0$. Since $g_\gamma(\cdot) = \sum_{i=1}^{p-1} g_\gamma^i(\cdot) + g_\gamma^p(\cdot)$, it is also strongly convex with the same parameter $\sigma_{g_\gamma^p} > 0$. □

A.5 The proof of Corollary 5.2: The Lipschitz gradient case.

From Lemma 5.1, we note that $\varphi_\gamma = -g_\gamma$ satisfies $\varphi_\gamma(\mathbf{y}) \geq \varphi(\hat{\mathbf{y}}) + \nabla\varphi_\gamma(\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}}) + (\sigma_g/2)\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2$, where $\sigma_g := (L_{f_p}+\gamma_0)^{-1}\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p) \leq (L_{f_p}+\gamma_k)^{-1}\lambda_{\min}(\mathbf{A}_p^T\mathbf{A}_p)$ for all $k \geq 0$ due to $\gamma_k \leq \gamma_0$. Using this inequality instead of the second inequality of (102) and $\gamma_{k+1} - \gamma_k = -\tau_k\gamma_{k+1}$, we obtain from (108) that

$$(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq H_{k+1}(\mathbf{w}) + \bar{\mathscr{T}}_{[4]}, \tag{117}$$

where

$$\begin{aligned}\bar{\mathscr{T}}_{[4]} &:= (\gamma_{k+1}/2)\big[\tau_k\|\mathbf{S}(\hat{\mathbf{x}}_k^\star - \mathbf{x}_c)\|_2^2 - (1-\tau_k)\tau_k\|\mathbf{S}(\bar{\mathbf{x}}_k^\star - \mathbf{x}_c)\|_2^2 + \sigma_g(1-\tau_k)\|\mathbf{S}(\bar{\mathbf{x}}_k^\star - \hat{\mathbf{x}}_k^\star)\|_2^2\big]\\ &\geq (\underline{\sigma}_g\gamma_{k+1}/2)\|\mathbf{S}(\hat{\mathbf{x}}_k^\star - \mathbf{x}_c) - (1-\tau_k)\mathbf{S}(\bar{\mathbf{x}}_k^\star - \mathbf{x}_c)\|_2^2.\end{aligned}$$

Here $\bar{\mathbf{x}}_k^\star := \mathbf{x}^\star_{\gamma_{k+1}}(\bar{\mathbf{y}}^k)$ and $\underline{\sigma}_g := \min\{\sigma_g, 1\} > 0$. We note that $\bar{\mathscr{T}}_{[4]} \geq 0$, taking the maximization both sides in (117) w.r.t. $\mathbf{w} \in \mathscr{W}$, we obtain $(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq G_{k+1}(\bar{\mathbf{w}}^{k+1}) + \psi_k$, where $\psi_k := (\underline{\sigma}_g\gamma_{k+1}/2)\|\mathbf{S}(\hat{\mathbf{x}}_k^\star - \mathbf{x}_c) - (1-\tau_k)\mathbf{S}(\bar{\mathbf{x}}_k^\star - \mathbf{x}_c)\|_2^2 \geq 0$. Finally, the proof of the estimates (53) in Corollary 5.2 can be done similarly as the proof of Theorem 4.1(c). □

A.6 The proof of Theorem 5.1: Inexact augmented Lagrangian method

We divide the prove into few steps as follows.
Step 1: Approximate smoothed gap function. Let us define an approximate gap function $G_{\gamma\beta}^\delta$ of the exact smoothed gap function $G_{\gamma\beta}$ in (25) as follows:

$$G_{\gamma\beta}^\delta(\bar{\mathbf{w}}) := \delta\text{-}\max_{\mathbf{w}\in\mathscr{W}}\left\{f(\bar{\mathbf{x}}) - f(\mathbf{x}) + F(\mathbf{w})^T(\bar{\mathbf{w}}-\mathbf{w}) - d_{\gamma\beta}(\mathbf{w})\right\}, \tag{118}$$

where the approximation only involves in $\mathbf{x}$ in the sense of (58), i.e.:

$$G_{\gamma\beta}(\bar{\mathbf{w}}) \leq G_{\gamma\beta}^\delta(\bar{\mathbf{w}}) + (\gamma/2)\delta^2. \tag{119}$$

Step 2: The first estimate of G_k. Let φ_γ be defined by (102), $\hat{\mathbf{x}}_k^\delta := \mathbf{x}_\gamma^\delta(\hat{\mathbf{y}}^k)$, $\varphi_\gamma^\delta(\mathbf{y}) := -f(\mathbf{x}_\gamma^\delta(\mathbf{y})) - (\mathbf{A}\mathbf{x}_\gamma^\delta(\mathbf{y}) - \mathbf{b})^T\mathbf{y} - \gamma d_b(\mathbf{S}\mathbf{x}_\gamma^\delta(\mathbf{y}), \mathbf{S}\mathbf{x}_c)$ and $\nabla\varphi_\gamma^\delta(\mathbf{y}) := \mathbf{b} - \mathbf{A}\mathbf{x}_\gamma^\delta(\mathbf{y})$. Then, by (58) we have

$$\varphi_\gamma(\mathbf{y}) - \varphi_\gamma^\delta(\mathbf{y}) \leq \gamma\delta^2/2 \quad\text{and}\quad \|\nabla\varphi_\gamma^\delta(\mathbf{y}) - \nabla\varphi_\gamma(\mathbf{y})\|_2 \leq \delta. \tag{120}$$

Since $\varphi_{\gamma_k}(\bar{\mathbf{y}}^k) \geq \varphi_{\gamma_k}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k)$ and $f(\hat{\mathbf{x}}_k^\delta) + (\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b})^T\mathbf{y} + (\gamma_k/2)\|\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b}\|_2^2 + \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\mathbf{y} - \hat{\mathbf{y}}^k) = 0$, it follows from (104) and $\beta_{k+1} = (1-\tau_k)\beta_k$ that

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq\ & (1-\tau_k)\big[\varphi_{\gamma_k}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k) + f(\bar{\mathbf{x}}^k) + (\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b})^T\mathbf{y}\big]\\ &+ \tau_k\big[f(\hat{\mathbf{x}}_k^\delta) + (\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b})^T\mathbf{y} + \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\mathbf{y} - \hat{\mathbf{y}}^k)\big]\\ &+ (\beta_{k+1}/2)\|\mathbf{y} - \bar{\mathbf{y}}_k^\star\|_2^2 - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 + (\tau_k\gamma_k/2)\|\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b}\|_2^2.\end{aligned}$$

Now, using (120), the third line $\bar{\mathbf{x}}^{k+1} = (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\tilde{\mathbf{x}}_{\gamma_k}^{\delta_k}(\hat{\mathbf{y}}^k)$ of (i1P2D), $\mathbf{u} := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\mathbf{y}$ and $\mathbf{u} - \hat{\mathbf{y}}^k = \tau_k(\mathbf{y} - \bar{\mathbf{y}}_k^\star)$, we can further estimate

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq\ & \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\mathbf{u} - \hat{\mathbf{y}}^k) + \frac{\beta_{k+1}}{2\tau_k^2}\|\mathbf{u} - \hat{\mathbf{y}}^k\|_2^2 + \frac{\tau_k\gamma_k}{2}\|\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b}\|_2^2 + f(\bar{\mathbf{x}}^{k+1})\\ &+ (1-\tau_k)\big[\nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k) - \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)\big]^T(\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - \frac{\beta_{k+1}}{2}\|\mathbf{y}\|_2^2. \qquad (121)\end{aligned}$$

Step 3: The second estimate of G_k. From the fourth line of (i1P2D) we have $\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\mathbf{u} - \hat{\mathbf{y}}^k) + \frac{\bar{L}^g}{2\gamma_k}\|\mathbf{u} - \hat{\mathbf{y}}^k\|_2^2 \geq \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) + \frac{\bar{L}^g}{2\gamma_k}\|\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k\|_2^2$. Using this inequality, $\bar{L}^g = 1$ and the condition $\beta_{k+1}\gamma_k \geq \bar{L}^g\tau_k^2 = \tau_k^2$ we can show that

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq\ & \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) + \frac{\bar{L}^g}{2\gamma_k}\|\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k\|_2^2 + f(\bar{\mathbf{x}}^{k+1}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y}\\ &- \frac{\beta_{k+1}}{2}\|\mathbf{y}\|_2^2 + \frac{\tau_k\gamma_k}{2}\|\mathbf{A}\hat{\mathbf{x}}_k^\delta - \mathbf{b}\|_2^2 + (1-\tau_k)\big[\nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k) - \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)\big]^T(\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k). \qquad (122)\end{aligned}$$

By using (120) and the first inequality of (102) we can write

$$\begin{aligned}\mathscr{T}_{[4]} :=\ & \varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) + \frac{\bar{L}^g}{2\gamma_k}\|\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k\|_2^2\\ \geq\ & \varphi_{\gamma_k}(\hat{\mathbf{y}}^k) + \nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k)^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) + \frac{\bar{L}^g}{2\gamma_k}\|\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k\|_2^2 + \big[\nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) - \nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k)\big]^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) - (\gamma_k\delta^2/2)\\ \geq\ & \varphi_{\gamma_k}(\bar{\mathbf{y}}^{k+1}) + \big[\nabla\varphi_{\gamma_k}^\delta(\hat{\mathbf{y}}^k) - \nabla\varphi_{\gamma_k}(\hat{\mathbf{y}}^k)\big]^T(\bar{\mathbf{y}}^{k+1} - \hat{\mathbf{y}}^k) - (\gamma_k\delta^2/2). \qquad (123)\end{aligned}$$

Substituting (123) into (122) and then using (120) and the definition of $\varphi_\gamma(\cdot)$ we get

$$\begin{aligned}(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \geq\ & \varphi_{\gamma_k}(\bar{\mathbf{y}}^{k+1}) + f(\bar{\mathbf{x}}^{k+1}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2\\ &- \delta_k\|(1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\hat{\mathbf{y}}^k - \bar{\mathbf{y}}^{k+1}\|_2 + \frac{\tau_k\gamma_k}{2}\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k} - \mathbf{b}\|_2^2 - (\gamma_k\delta_k^2/2)\\ \geq\ & f(\bar{\mathbf{x}}^{k+1}) - f(\mathbf{x}) + (\mathbf{A}\bar{\mathbf{x}}^{k+1} - \mathbf{b})^T\mathbf{y} - (\mathbf{A}\mathbf{x} - \mathbf{b})^T\bar{\mathbf{y}}^{k+1} - (\beta_{k+1}/2)\|\mathbf{y}\|_2^2 - (\gamma_k/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 - \mathscr{T}_{[5]}\\ \geq\ & H_{k+1}(\bar{\mathbf{w}}^{k+1}) - \mathscr{T}_{[5]}, \qquad (124)\end{aligned}$$

provided that $\gamma_{k+1} \geq \gamma_k$, where $\mathscr{T}_{[5]} := \delta_k\|(1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\hat{\mathbf{y}}^k - \bar{\mathbf{y}}^{k+1}\|_2 + (\gamma_k\delta_k^2)/2 - (\tau_k\gamma_k/2)\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k} - \mathbf{b}\|_2^2$.
Step 4: Simplify $\mathscr{T}_{[5]}$ to obtain (27). Using the definition of $\hat{\mathbf{y}}^k$ and $\bar{\mathbf{y}}^{k+1}$ we can further estimate $\mathscr{T}_{[5]}$ as

$$\mathscr{T}_{[5]} := (1-\tau_k)\tau_k\delta_k\|\bar{\mathbf{y}}^k - \hat{\mathbf{y}}^k\|_2 + (\gamma_k\delta_k^2)/2 + \gamma_k\delta_k\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k} - \mathbf{b}\|_2 - (\tau_k\gamma_k/2)\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k} - \mathbf{b}\|_2^2. \qquad (125)$$

Taking the maximization of (124) over $\mathbf{w} \in \mathscr{W}$ we finally obtain

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) \leq (1-\tau_k)G_k(\bar{\mathbf{w}}^k) - \psi_k, \qquad (126)$$

where $\psi_k := (\tau_k\gamma_0/2)\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k}-\mathbf{b}\|_2^2-(1-\tau_k)\tau_k\delta_k\|\bar{\mathbf{y}}^k-\bar{\mathbf{y}}_k^\star\|_2-\gamma_0\delta_k\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k}-\mathbf{b}\|_2-(\gamma_0\delta_k^2)/2$.
Step 5: Prove (60). We note that $\|\mathbf{A}\hat{\mathbf{x}}_k^{\delta_k}-\mathbf{b}\|_2 \le D_{\mathscr{Z}}^{\mathbf{A}}$ and $\gamma_0 = \bar{L}^g = 1$, which lead to $\psi_k \ge -q_k\delta_k$, where $q_k := (1-\tau_k)\tau_k\|\bar{\mathbf{y}}^k-\bar{\mathbf{y}}_k^\star\|_2+(D_{\mathscr{Z}}^{\mathbf{A}}+1)/2$. In this case (126) leads to $G_{k+1}(\bar{\mathbf{w}}^{k+1}) \le (1-\tau_k)G_k(\bar{\mathbf{w}}^k)+q_k\delta_k$. Therefore, if we choose δ_k so that $q_k\delta_k \le q_{k-1}\delta_{k-1}$. Then, by induction and $\prod_{i=0}^k(1-\tau_k) \le \frac{4}{(k+2)^2}$ due to Lemma 4.4, the last estimate leads to

$$G_k(\bar{\mathbf{w}}^k) \le \omega_k G_0(\bar{\mathbf{w}}^0)+q_0\delta_0+4\sum_{j=1}^{k-1}\frac{q_j\delta_j}{(j+1)^2} \le \omega_k G_0(\bar{\mathbf{w}}^0)+4q_0\delta_0\zeta(2). \tag{127}$$

Here $\zeta(s) := \sum_{j=1}^\infty j^{-s}$ is the zeta-function. We note that the starting point $\bar{\mathbf{w}}^0$ is also computed up to the accuracy δ_0, i.e. $G_0(\bar{\mathbf{w}}^0) \le (\gamma_0\delta_0^2/2)$ and $\zeta(2) < 1.64494$. Plugging these into (127) we have $G_k(\bar{\mathbf{w}}^k) \le 7q_0\delta_0$. Combining this inequality and Lemma 3.3 we obtain the second estimate of (60). Finally, by using the bound $G_k(\bar{\mathbf{w}}^k) \le 7q_0\delta_0$, it follows from (111) that $\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|_2 \le \frac{4}{(k+1)^2}\left[2D_{\mathscr{Y}}^\star+\sqrt{\frac{14q_0\delta_0}{(k+1)^2}}\right]$, which is the first estimate in (60). □

A.7 The proof of Corollary 6.2: PADMM variant.

Since γ_k is fixed at $\gamma_k = \gamma_0 > 0$ and $d_b(\mathbf{Sx},\mathbf{Sx}_c) := \frac{1}{2}\|\mathbf{x}-\mathbf{s}^k\|_2^2$ where $\mathbf{s}^k := (\mathbf{g}_1^k,\mathbf{g}_2^k)$. From the proof (108) of Lemma 4.2, we can estimate

$$(1-\tau_k)G_k(\bar{\mathbf{w}}^k) \ge G_{k+1}(\bar{\mathbf{w}}^{k+1})+\gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}_{k+1}^\star,\mathbf{S}\hat{\mathbf{x}}_k^\star)-\gamma_0 d_b(\mathbf{S}\bar{\mathbf{x}}_k^\star,\mathbf{S}\hat{\mathbf{x}}_{k-1}^\star),$$

where $\bar{\mathbf{x}}_k^\star := \mathbf{x}_{\gamma_0}^\star(\bar{\mathbf{y}}^k)$. By induction, it follows from the previous inequality that

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) \le \omega_k G_0(\bar{\mathbf{w}}^0)-\Psi_k, \text{ where } \Psi_k = \gamma_0[\psi_0+\sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)[d_j-d_{j-1}],$$

where $d_k := d_b(\mathbf{S}\bar{\mathbf{x}}_{k+1}^\star,\mathbf{S}\hat{\mathbf{x}}_k^\star)$. Since $\prod_{l=0}^{j-1}(1-\tau_l) \ge (j+1)^{-2}$ and $d_j-d_{j+1} \ge -2D_{\mathscr{Z}}^{\mathbf{S}}$, we can show that $\Psi_k \ge \gamma_0[\Psi_0-2D_{\mathscr{Z}}^{\mathbf{S}}\zeta(2)] \ge -4\gamma_0 D_{\mathscr{Z}}^{\mathbf{S}}$. Consequently, we obtain the bound in (65). □

A.8 The proof of Corollary 6.1: The convergence of the new ADMM variant.

Similarly to the proof of Corollary 6.2, we can show that

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) \le \omega_k G_0(\bar{\mathbf{w}}^0)-\Psi_k, \ \Psi_k = \gamma_0[\psi_0+\sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)[d_j-d_{j-1}],$$

and $d_k := (1/2)\|\mathbf{A}_1\mathbf{x}_{\gamma_k}^\star(\hat{\mathbf{y}}^k)+\mathbf{A}_2\mathbf{x}_{\gamma_{k-1}}^\star(\hat{\mathbf{y}}^{k-1})-\mathbf{b}\|_2^2+(1/2)\|\mathbf{A}_1\mathbf{x}_{\gamma_k}^\star(\hat{\mathbf{y}}^k)+\mathbf{A}_2\mathbf{x}_{\gamma_k}^\star(\hat{\mathbf{y}}^k)-\mathbf{b}\|_2^2$. Then, we can estimate $\Psi_k \ge -4\gamma_0 D_{\mathscr{Z}}^{\mathbf{A}}$. Consequently, we obtain the bound in (63). □.

References

[1] A. Auslender. *Optimisation: Méthodes Numériques*. Masson, Paris, 1976.

[2] H.H. Bauschke and P. Combettes. *Convex analysis and monotone operators theory in Hilbert spaces*. Springer-Verlag, 2011.

[3] A. Beck and D. Pan. On the solution of the GPS localization and circle fitting problems. *SIAM J. Optim.*, 22(1):108–134, 2012.

[4] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[5] A. Beck and M. Teboulle. A fast dual proximal gradient algorithm for convex minimization and applications. *Oper. Res. Letter*, 42(1):1–6, 2014.

[6] S. Becker, J. Bobin, and E.J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Science*, 4(1):1–39, 2011.

[7] A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: Pivotal recovery of sparse signals via conic programming. *Biometrika*, 94(4):791–806, 2011.

[8] E. Van Den Berg and M. P. Friedlander. Probing the Pareto frontier for basic pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[9] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and Ö. Yılmaz. Sparco: A testing framework for sparse reconstruction. Tech. Report TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver, October 2007.

[10] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific, 1996 (Optimization and Neural Computation Series).

[11] D.P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[14] E.J. Candés, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[15] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[16] S. H. Chan, R. Khoshabeh, K.B. Gibson, P. E. Gill, and T.Q. Nguyen. An Augmented Lagrangian Method for Total Variation Video Restoration. *IEEE Trans. Image Processing*, 20(11):3097–3111, 2011.

[17] V. Chandrasekaran, P.A. Parrilo, and A.S. Willsky. Latent variable graphical model selection via convex optimization. *The annals of Statistics*, 40(4):1935–1967, 2012.

[18] V. Chandrasekaranm, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[19] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011.

[20] G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Math. Program.*, 64:81–101, 1994.

[21] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4:1168–1200, 2005.

[22] L. Condat. A primaldual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory and Appl.*, xx:1–20, 2012.

[23] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Tech. Report No. TR12-14, Rice University CAAM, 2012.

[24] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[25] D.L. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 25(4):1289–1306, 2006.

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

[27] J. E. Esser. *Primal-dual algorithm for convex models and applications to image restoration, registration and nonlocal inpainting*. Phd. thesis, University of California, Los Angeles, Los Angeles, USA, 2010.

[28] F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*, vol. 1–2. Springer-Verlag, 2003.

[29] M. Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Math. Program.*, 53:99–110, 1992.

[30] D. Goldfarb and S. Ma. Fast alternating linearization methods of minimization of the sum of two convex functions. *Math. Program., Ser. A*, pages 1–34, 2012.

[31] T. Goldstein, E. Esser, and R. Baraniuk. Adaptive Primal-Dual Hybrid Gradient Methods for Saddle Point Problems. *Tech. Report*, 1–26, 2013 (`http://arxiv.org/pdf/1305.0546v1.pdf`).

[32] T. Goldstein, B. ODonoghue, and S. Setzer. Fast Alternating Direction Optimization Methods. Tech. Report, Department of Mathematics, University of California, Los Angeles, USA, May 2012.

[33] M. Grant. *Disciplined Convex Programming*. PhD thesis, Stanford University, 2004.

[34] A. Hamdi. Decomposition for structured convex programs with smooth multiplier methods. *Applied Mathematics and Computation*, 169:218–241, 2005.

[35] A. Hamdi. Two-level primal-dual proximal decomposition technique to solve large-scale optimization problems. *Appl. Math. Comput.*, 160:921–938, 2005.

[36] B. He and X. Yuan. Convergence analysis of primal-dual algorithms for saddle-point problem: from contraction perspective. *SIAM J. Imaging Sciences*, 5:119–149, 2012.

[37] B. He and X. Yuan. On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. Tech. Report, Hong Kong Baptist University, pp. 1–9, 2012.

[38] B.S. He and X.M. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50:700–709, 2012.

[39] G. Lan and R.D.C. Monteiro. Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Tech. Report*, University of Florida, 2013.

[40] S. Lefkimmiatis and M. Unser. Poisson Image Reconstruction with Hessian Schatten-Norm Regularization. *EEE Trans. Image Processing*, 22(11):4314–4327, 2013.

[41] Z. Lin, M. Chen, L. Wu, and Y. Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *UIUC Tech. Report*, No. UILU-ENG-09-2215, 2009.

[42] S. Ma, L. Xue, and H. Zou. Alternating direction methods for latent variable gaussian graphical model selection. *Neural Computation*, 25(8):2172–2198, 2013.

[43] M. B McCoy, V. Cevher, Q. Tran-Dinh, A. Asaei, and L. Baldassarre. Convexity in source separation: Models, geometry, and algorithms. *IEEE Signal Processing Magazine*, 31(3):87–95, 2014.

[44] I. Necoara and J.A.K. Suykens. Applications of a smoothing technique to decomposition in convex optimization. *IEEE Trans. Automatic control*, 53(11):2674–2679, 2008.

[45] V. Nedelcu, I. Necoara, and Q. Tran-Dinh. Computational Complexity of Inexact Gradient Augmented Lagrangian Methods: Application to Constrained MPC. *SIAM J. Optim. Control*, (partially accepted), 2014.

[46] A. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17(1):191–234, 2008.

[47] A. Nemirovskii. Prox-method with rate of convergence $\mathcal{O}(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Op*, 15(1):229–251, 2004.

[48] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, Vol. 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[49] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optimization*, 16(1):235–249, 2005.

[50] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[51] Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Program.*, 109(2–3):319–344, 2007.

[52] Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.

[53] Y. Nesterov. Gradient methods for minimizing composite objective function. *Mathematical Programming*, 140(1):125–161, 2013.

[54] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[55] B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 2012 (to appear).

[56] H. Ouyang, N. He, Long Q. Tran, and A. Gray. Stochastic alternating direction method of multipliers. *JMLR W&CP*, 28:80–88, 2013.

[57] Y. Ouyang, Y. Chen, G. LanG. Lan., and E. JR. Pasiliao. An accelerated linearized alternating direction method of multiplier. *Tech*, 2014.

[58] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[59] R. A. Polyak, J. Costa, and J. Neyshabouri. Dual fast projected gradient method for quadratic programming. *Optimization Letters*, 7(4):631–645, 2013.

[60] R. T. Rockafellar. *Convex Analysis*, Vol. 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

[61] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathemathics of Operations Research*, 1:97–116, 1976.

[62] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control and Optim.*, 14:877–898, 1976.

[63] R.T. Rockafellar. *Convexity and Duality in Optimization*, chapter Monotropic Programming: A generalization of linear programming and network programming., pp. 10–036. Springer-Verlag, 1985.

[64] R. Shefi and M. Teboulle. Rate of Convergence Analysis of Decomposition Methods Based on the Proximal Method of Multipliers for Convex Minimization. *SIAM J. Optim.*, 24(1):269–297, 2014.

[65] K.-Ch. Toh, M.J. Todd, and R.H. Tütüncü. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, Version 4.0. *Tech. Report*, NUS Singapore, 2010.

[66] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *Tech. Report., LIONS, EPFL*, pages 1–42, 2013.

[67] Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Compt. Optim. Appl.*, 55(1):75–111, 2013.

[68] P. Tseng. Applications of splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.*, 29:119–138, 1991.

[69] M. J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *nnual Review of Statistics and its Applications*, 1:233–253, 2014.

[70] H. Wang and A. Banerjee. Bregman Alternating Direction Method of Multipliers. pp. 1–18, 2013 (`http://arxiv.org/pdf/1306.3203v1.pdf`).

[71] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1 -problems in compressive sensing. *SIAM J. Scientific Computing*, 33(1–2):250–278, 2011.