An Adaptive Proximal ADMM for Nonconvex Linearly Constrained Composite Programs *

Leandro Farias Maia[†] David H. Gutman[‡] Renato D.C. Monteiro[§]
Gilson N. Silva[¶]

July 12, 2024
(1st revision: June 30, 2025; 2nd revision: Jan 16, 2026; 3rd revision: April 10, 2026)

Abstract

This paper develops an adaptive proximal alternating direction method of multipliers (ADMM) for solving linearly constrained, composite optimization problems under the assumption that the smooth component of the objective is weakly convex, while the non-smooth component is a convex block-separable function with compact domain. The proposed method is adaptive to all problem parameters, including smoothness and weak convexity constants, and allows each of its block proximal subproblems to be inexactly solved. Each iteration of our adaptive proximal ADMM consists of two steps: the sequential solution of each block proximal subproblem; and adaptive tests to decide whether to perform a full Lagrange multiplier and/or penalty parameter update(s). Without any rank assumptions on the constraint matrices, it is shown that the adaptive proximal ADMM obtains an approximate first-order stationary point of the constrained problem in a number of iterations that matches the state-of-the-art complexity for the class of proximal ADMM's. The three proof-of-concept numerical experiments that conclude the paper suggest our adaptive proximal ADMM enjoys significant computational benefits.

Keywords: proximal ADMM, nonseparable, nonconvex composite optimization, iteration-complexity, augmented Lagrangian function

1 Introduction

This paper develops an adaptive proximal alternating direction method of multipliers, called Adapt-ADMM, for solving the linearly constrained, smooth, weakly convex, composite optimization problem

$$\phi^* = \min_{y\in\mathbb{R}^n} \left\{\phi(y) := f(y) + h(y) : Ay = b\right\}, \tag{1}$$

where $A : \mathbb{R}^n \to \mathbb{R}^l$ is a linear operator, $b \in \mathbb{R}^l$ is a vector in the image of A, h is a proper closed convex function which is Lipschitz continuous on its compact domain and, for some positive integer B (the number of blocks) and positive integer vector $(n_1, \ldots, n_t)$ such that $n = \sum_{t=1}^{B} n_t$, has the blockwise representation $h(y) = \sum_{t=1}^{B} h_t(y_t)$ for every $y = (y_1, \ldots, y_B) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_B}$, and f is a real-valued weakly convex differentiable function on the domain of h (assumed compact), whose gradient satisfies a blockwise Lipschitz

*Funding: The first and second authors are supported by the NSF grant #2410328. The third author has been partially supported by AFOSR Grants FA9550-22-1-0088 and FA9550-25-1-0131. The last author has been supported by CNPq Grants 401864/2022-7 and 306593/2022-0

[†]School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR, 97331. farias-maia@gmail.com

[‡]Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, 77843. dhgutman@gmail.com

[§]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. monteiro@isye.gatech.edu

[¶]Department of Mathematics, Federal University of Piauí, Teresina, PI, 64049-550. gilson.silva@ufpi.edu.br

condition. Thus, in terms of this blockwise representation, $f(y)$, $h(y)$, and Ay, can be written as

$$f(y) = f(y_1, \ldots, y_B), \quad Ay = \sum_{t=1}^{B} A_t y_t, \quad h(y) = \sum_{t=1}^{B} h_t(y_t), \tag{2}$$

where, for each $t \in \{1, \ldots, B\}$, h_t is a proper closed convex function with compact domain and $A_t : \mathbb{R}^{n_t} \to \mathbb{R}^l$ is a linear operator.

The goal in this paper is to find a (ρ, η)-stationary solution of (1)-(2), i.e., a quadruple $(\hat{x}, \hat{p}, \hat{u}, \hat{\varepsilon}) \in (\mathrm{dom}\, h) \times A(\mathbb{R}^n) \times \mathbb{R}^l \times \mathbb{R}_+$ satisfying

$$\hat{u} \in \nabla f(\hat{x}) + \partial_{\hat{\varepsilon}} h(\hat{x}) + A^* \hat{p}, \quad \sqrt{\|\hat{u}\|^2 + \hat{\varepsilon}} \le \rho, \quad \|A\hat{x} - b\| \le \eta, \tag{3}$$

where $(\rho, \eta) \in \mathbb{R}^2_{++}$ is a given tolerance pair and the $\hat{\varepsilon}$-subdifferential of h is defined in Subsection 1.2 below.

A popular primal-dual algorithmic framework for solving problem (1) that takes advantage of its block structure (2) is the proximal ADMM, which is based on the augmented Lagrangian (AL) function,

$$\mathcal{L}_c(y; p) := \phi(y) + \langle p, Ay - b \rangle + \frac{c}{2} \|Ay - b\|^2, \tag{4}$$

where $c > 0$ is a penalty parameter. Given $(\tilde{y}^{k-1}, \tilde{q}^{k-1}, c_{k-1})$, the proximal ADMM finds the next triple $(\tilde{y}^k, \tilde{q}^k, c_k)$ as follows. Starting from $\tilde{y}^{k-1}$, it first performs ℓ_k block inexact proximal point (BIPP) iterations applied to $\mathcal{L}_{c_{k-1}}(\cdot\,; \tilde{q}^{k-1})$ to obtain $\tilde{y}_k$ where ℓ_k is a positive integer. Next, it performs a Lagrange multiplier update according to

$$\tilde{q}^k = (1 - \theta)\Big[\tilde{q}^{k-1} + \chi c_k \left(A\tilde{y}^k - b\right)\Big], \tag{5}$$

where $\theta \in [0, 1)$ is a dampening parameter and χ is a positive relaxation parameter, and chooses a scalar $c_k \ge c_{k-1}$ as the next penalty parameter.

We now formally describe how a proximal ADMM obtains $\tilde{y}^k$ from $\tilde{y}^{k-1}$. It sets $z^0 = \tilde{y}^{k-1}$, and for some positive integer ℓ_k, it performs a BIPP iteration from z^{j-1} to obtain z^j for every $j = 1, \ldots \ell_k$, and finally sets $\tilde{y}^k = z^{\ell_k}$. The BIPP iteration to obtain z^j from z^{j-1} consists of inexactly solving, sequentially from $t = 1$ to $t = B$, the t-th block proximal AL subproblem with prox stepsize λ_t

$$z_t^j \approx \mathrm{argmin}_{\,u_t \in \mathbb{R}^{n_t}} \left\{ \lambda_t \mathcal{L}_{c_{k-1}}(z^j_{<t}, u_t, z^{j-1}_{>t}; \tilde{q}^{k-1}) + \frac{1}{2}\|u_t - z_t^{j-1}\|^2 \right\}, \tag{6}$$

and finally setting $\tilde{y}^k = z^{\ell_k}$.

The recent publication [31] proposes a version of a proximal ADMM for solving (1)-(2) which assumes that $\ell_k = 1$, $\lambda_1 = \cdots = \lambda_B$, and $(\chi, \theta) \in (0, 1]^2$ satisfies

$$2\chi B(2 - \theta)(1 - \theta) \le \theta^2, \tag{7}$$

and hence that $\theta = 0$ is not allowed in [31].

One of the main contributions of [31] is that its convergence guarantees do not require *the last block condition*, $\mathrm{Im}(A_B) \supseteq \{b\} \cup \mathrm{Im}(A_1) \cup \ldots \cup \mathrm{Im}(A_{B-1})$ and $h_B \equiv 0$, that hinders many instances of proximal ADMM, see [9, 19, 60, 64]. However, the main drawbacks of the proximal ADMM of [31] include: (i) the strong assumption (7) on (χ, θ); (ii) subproblem (6) must be solved exactly; (iii) the stepsize parameter λ is conservative and requires the knowledge of f's weak convexity parameter; (iv) it (conservatively) updates the Lagrange multiplier after each primal update cycle (i.e., $\ell_k = 1$); (v) its iteration-complexity has a high dependence on the number of blocks B, namely, $\mathcal{O}(B^8)$; (vi) its iteration-complexity bound depends linearly on θ^{-1}, and hence grows to infinity as θ approaches zero. Paper [31] also presents computational results comparing its proximal ADMM with a more practical variant where (θ, χ), instead of satisfying (7), is set to $(0, 1)$. Intriguingly, this $(\theta, \chi) = (0, 1)$ regime substantially outperforms the theoretical regime of (7) in the provided computational experiments. No convergence analysis for the $(\theta, \chi) = (0, 1)$ regime is forwarded in [31]. Thus, [31] leaves open the tantalizing question of whether the convergence of proximal ADMM with $(\theta, \chi) = (0, 1)$ can be theoretically secured.

Contributions: This work partially addresses the convergence analysis issue raised above by studying a *completely parameter-free* proximal ADMM, with $(\theta, \chi) = (0, 1)$ and ℓ_k adaptively chosen, called Adapt-ADMM. Rather than making the conservative determination that $\ell_k = 1$, the studied adaptive method ensures the dual updates are committed as frequently as possible. It is shown that Adapt-ADMM finds a (ρ, η)-stationary solution in $\mathcal{O}(B \max\{\rho^{-3}, \eta^{-3}\})$ iterations. Adapt-ADMM also exhibits the following additional features:

- Similar to the proximal ADMM of [31], its complexity is established without assuming that the *last block condition* holds.
- Compared to the $\mathcal{O}(B^8 \max\{\rho^{-3}, \eta^{-3}\})$ iteration-complexity of the proximal ADMM of [31], the one for Adapt-ADMM vastly *improves the dependence on* B.
- Adapt-ADMM uses a scheme that adaptively computes *variable block prox stepsizes*, instead of constant ones whose expressions depend on the weakly convex parameters of f as in the proximal ADMM of [31]. Specifically, while the method of [31] chooses $\lambda_1 = \ldots = \lambda_B \in (0, 1/(2\bar{m})]$ where $\bar{m}$ is a weakly convex parameter for $f(y)$ relative to the whole y, VP-ADMM adaptively generates possibly distinct λ_t's that are larger than $1/(2m_t)$ (and hence $1/(2\bar{m})$) where m_t is the weakly convex parameter of f relative to its t-th block y_t. Thus, Adapt-ADMM allows some of (or all) the subproblems (6) to be non-convex.
- Adapt-ADMM is also adaptive to Lipschitz parameters.
- In contrast to the proximal ADMM in [31], Adapt-ADMM allows the block proximal subproblems (6) to be either exactly or *inexactly* solved.

Related Works: ADMM methods with $B = 1$ are well-known to be equivalent to augmented Lagrangian methods. Several references have studied augmented Lagrangian and proximal augmented Lagrangian methods in the convex (see e.g., [1, 2, 36, 37, 38, 40, 48, 53, 62]) and nonconvex (see e.g. [5, 6, 20, 25, 30, 34, 35, 41, 59, 63, 64, 65]) settings. Moreover, ADMMs and proximal ADMMs in the convex setting have also been broadly studied in the literature (see e.g. [5, 7, 10, 11, 12, 13, 14, 16, 17, 18, 46, 54, 57]). So from now on, we just discuss proximal ADMM variants where f is nonconvex and $B > 1$.

A discussion of the existent literature on nonconvex proximal ADMM is best framed by dividing it into two different corpora: those papers that assume the last block condition and those that do not. Under the *last block condition*, the iteration-complexity established is $\mathcal{O}(\varepsilon^{-2})$, where $\varepsilon := \min\{\rho, \eta\}$. Specifically, [9, 19, 60, 61] introduce proximal ADMM approaches assuming $B = 2$, while [26, 27, 42, 43] present (possibly linearized) proximal ADMMs assuming $B \geq 2$. A first step towards removing the last block condition was made by [27] which proposes an ADMM-type method applied to a penalty reformulation of (1)-(2) that artificially satisfies the last block condition. This method possesses an $\mathcal{O}(\varepsilon^{-6})$ iteration-complexity bound.

On the other hand, development of ADMM-type methods directly applicable to (1)-(2) is considerably more challenging and only a few works addressing this topic have surfaced. In addition to [27], earlier contributions to this topic were obtained in [24, 59, 64]. More specifically, [24, 64] develop a novel small stepsize ADMM-type method without establishing its complexity. Finally, [59] considers an interesting but unorthodox negative stepsize for its Lagrange multiplier update, that sets it outside the ADMM paradigm, and thus justifies its qualified moniker, "scaled dual descent ADMM".

1.1 Structure of the Paper

This subsection outlines this article's structure. Subsection 1.2 briefly lays out the basic definitions and notation used throughout. Section 2 introduces a notion of an inexact solution of Adapt-ADMM's foundational block proximal subproblem (6) and discusses efficient ways to find said solutions.

The ADMM variants considered in Sections 3 to 5 assume that the weak convexity parameters m_t's are known and use them to compute their prox stepsize, which is kept *constant* throughout its execution. Specifically, Section 3 presents a static (i.e., with fixed penalty parameter) ADMM variant, FP-ADMM, and states its main result, Theorem 3.2, governing its iteration-complexity. Section 4 provides the detailed proof of Theorem 3.2 and presents all supporting technical lemmas. Section 5 presents a non-static (i.e., with variable penalty parameter) ADMM variant, namely VP-ADMM, and establishes its iteration-complexity in Theorem 5.2.

Section 6 presents the centerpiece algorithm of this work, Adapt-ADMM, an *adaptive* prox stepsize version of VP-ADMM that requires no knowledge of the weak convexity parameters m_t's.

Section 7 presents proof-of-concept numerical experiments that display the efficiency of Adapt-ADMM for three different problem classes. Section 8 gives some concluding remarks that suggest further research directions. Appendix A presents some technical results on convexity and linear algebra, while Appendix B describes an adaptive accelerated gradient method and its main properties. Finally, Appendix C discusses some technical results on the inexact solution notion adopted in this work and its connection to directional derivatives.

1.2 Notation, Definitions, and Basic Facts

This subsection lists the elementary notation deployed throughout the paper. Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of non-negative and positive real numbers, respectively. We assume that the n-dimensional Euclidean space, $\mathbb{R}^n$, is equipped with an inner product, $\langle \cdot\,, \cdot \rangle$.

The norm induced by $\langle \cdot\,, \cdot \rangle$ is denoted by $\|\cdot\|$. Let $\mathbb{R}^n_{++}$ and $\mathbb{R}^n_+$ denote the set of vectors in $\mathbb{R}^n$ with positive and non-negative entries, respectively. The smallest positive singular value of a nonzero linear operator $Q : \mathbb{R}^n \to \mathbb{R}^l$ is denoted ν_Q^+ and its operator norm is $\|Q\| := \sup\{\|Q(w)\| : \|w\| = 1\}$. If S is a symmetric and positive definite matrix, the norm induced by S on $\mathbb{R}^n$, denoted by $\|\cdot\|_S$, is defined as $\|\cdot\|_S = \langle \cdot\,, S(\cdot)\rangle^{1/2}$. For $x = (x_1, \ldots, x_B) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_B}$, we define the aggregated quantities

$$x_{<t} := (x_1, \ldots, x_{t-1}), \quad x_{>t} := (x_{t+1}, \ldots, x_B), \quad x_{\le t} := (x_{<t}, x_t), \quad x_{\ge t} := (x_t, x_{>t}). \tag{8}$$

Moreover, for $a \in \mathbb{R}^B$, we define

$$\min(a) = \min_{1\le t\le B} a_t \quad \text{and} \quad \max(a) = \max_{1\le t\le B} a_t. \tag{9}$$

For a given closed, convex set $Z \subset \mathbb{R}^n$, we let ∂Z designate its boundary. The distance of a point $z \in \mathbb{R}^n$ to Z, measured in terms of $\|\cdot\|$, is denoted $\operatorname{dist}(z, Z)$. The indicator function of Z, denoted by δ_Z, is defined by $\delta_Z(z) = 0$ if $z \in Z$, and $\delta_Z(z) = +\infty$ otherwise.

For a given function $g : \mathbb{R}^n \to (-\infty, \infty]$, let $\operatorname{dom} g := \{x \in \mathbb{R}^n : g(x) < +\infty\}$ denotes the effective domain of g. We say that g is proper if $\operatorname{dom} g \ne \emptyset$. The set of all lower semi-continuous proper convex functions defined in $\mathbb{R}^n$ is denoted by $\overline{\operatorname{Conv}}\,(\mathbb{R}^n)$. For $\varepsilon \ge 0$, the ε-subdifferential of $g \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$ at $z \in \operatorname{dom} g$ is

$$\partial_\varepsilon g(z) := \{w \in \mathbb{R}^n \ : \ g(\tilde{z}) \ge g(z) + \langle w\,, \tilde{z} - z\rangle - \varepsilon, \forall \tilde{z} \in \mathbb{R}^n\}. \tag{10}$$

When $\varepsilon = 0$, the ε-subdifferential recovers the classical subdifferential, $\partial g(\cdot) := \partial_0 g(\cdot)$. It is well-known (see [23, Prop. 1.3.1 of Ch. XI]) that for any $\beta > 0$ and $g \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$,

$$\partial_\varepsilon(\beta g)(\cdot) = \beta \partial_{(\varepsilon/\beta)} g(\cdot). \tag{11}$$

Moreover, if $h_i \in \overline{\operatorname{Conv}}\,(\mathbb{R}^{n_i})$ for $i = 1, \ldots, B$ and $h(y) := \sum_{t=1}^B h_t(y_t)$ for any $y = (y_1, \ldots, y_B)$, then we have (see [23, Remark 3.1.5 of Ch. XI])

$$\partial_\varepsilon h(y) = \cup\{\partial_{\varepsilon_1} h_1(y_1) \times \ldots \times \partial_{\varepsilon_B} h_B(y_B) : \varepsilon_t \ge 0, \ \ \varepsilon_1 + \cdots + \varepsilon_B \le \varepsilon\}. \tag{12}$$

2 Assumptions and an Inexact Solution Concept

This section contains two subsections. The first one details a few mild technical assumptions imposed on the main problem (1)-(2). The second one introduces a notion of an inexact stationary point for the block proximal subproblem (6) along with an efficient method for finding such points.

2.1 Assumptions for Problem (1)-(2)

The main problem of interest in this paper is problem (1) with the block structure as in (2). It is assumed that vector $b \in \mathbb{R}^l$, linear operator $A : \mathbb{R}^n \to \mathbb{R}^l$, and functions $f : \mathbb{R}^n \to (-\infty, \infty]$ and $h_t : \mathbb{R}^{n_t} \to (-\infty, \infty]$ for $t = 1, \ldots, B$, satisfy the following conditions:

(A1) $h(\cdot)$ as in (2) satisfies the following properties:

- for every $t = 1, \ldots, B$, $h_t(\cdot) \in \overline{\operatorname{Conv}}\,(\mathbb{R}^{n_t})$ is prox friendly (i.e., its proximal operator is easily computable) and its domain $\mathcal{H}_t$ is compact;

- there exists $M_h \geq 0$ such that $h(\cdot)$ restricted to $\mathcal{H} := \mathcal{H}_1 \times \cdots \times \mathcal{H}_B$ is M_h-Lipschitz continuous;

(A2) for some $m = (m_1, \ldots, m_B) \in \mathbb{R}^B_+$, function f is block m-weakly convex, i.e., for every $t \in \{1, \ldots, B\}$,

$$f(x_{<t}, \cdot, x_{>t}) + \delta_{\mathcal{H}_t}(\cdot) + \frac{m_t}{2}\|\cdot\|^2 \text{ is convex for all } x \in \mathcal{H};$$

(A3) f is differentiable on $\mathcal{H}$ and, for every $t \in \{1, \ldots, B-1\}$, there exists $L_{>t} \geq 0$ such that

$$\|\nabla_t f(x_{\leq t}, \tilde{x}_{>t}) - \nabla_t f(x_{\leq t}, x_{>t})\| \leq (L_{>t})\|\tilde{x}_{>t} - x_{>t}\| \qquad \forall\, x, \tilde{x} \in \mathcal{H}, \tag{13}$$

where $\nabla_t f(\cdot)$ denotes the t-th block component of the whole gradient $\nabla f(\cdot)$;

(A4) A is nonzero and there exists $\bar{\mathrm{x}} \in \{x \in \mathcal{H} : Ax = b\} \neq \emptyset$ such that $\bar{d} := \mathrm{dist}(\bar{\mathrm{x}}, \partial\mathcal{H}) > 0$.

We now make some remarks about the above assumptions. First, since $\mathcal{H}$ is compact by (A1), the scalars

$$D_h := \sup_{z \in \mathcal{H}} \|z - \bar{x}\|, \quad \widehat{\nabla}_f := \sup_{u \in \mathcal{H}} \|\nabla f(u)\|, \quad \underline{\phi} := \inf_{u \in \mathcal{H}} \phi(u), \quad \overline{\phi} := \sup_{u \in \mathcal{H}} \phi(u), \tag{14}$$

are bounded. Second, (A2) allows m_t to be zero for some or all $t \in \{1, \ldots, B\}$. Finally, it follows from (A4) that $\bar{\mathrm{x}}$ is a Slater point for (1) in the strong sense that $\bar{\mathrm{x}} \in \mathrm{int}\,(\mathcal{H})$ and $A\bar{\mathrm{x}} = b$.

2.2 An Inexact Solution Concept for (6)

This subsection introduces our notion (Definition 2.1) of an inexact solution of the block proximal AL subproblem (6). To cleanly frame this solution concept, observe that (6) can be cast in the form

$$\min\{\psi(z) := \psi_{\mathrm{s}}(z) + \psi_{\mathrm{n}}(z) : z \in \mathbb{R}^n\}, \tag{15}$$

where

$$\psi_{\mathrm{s}}(\cdot) = \lambda_t \hat{\mathcal{L}}_c(y^i_{<t}, \cdot, y^{i-1}_{>t}; \tilde{q}^{k-1}) + \frac{1}{2}\|\cdot - y^{i-1}_t\|^2, \quad \psi_{\mathrm{n}}(\cdot) = \lambda_t h_t(\cdot), \tag{16}$$

and, for every $p \in \mathbb{R}^l$, $\hat{\mathcal{L}}_c(\cdot\,; p)$ denotes the smooth part of (4), i.e.,

$$\hat{\mathcal{L}}_c(y; p) := f(y) + \langle p, Ay - b\rangle + \frac{c}{2}\|Ay - b\|^2. \tag{17}$$

Hence, to describe a notion of an inexact solution for (6), it suffices to do so in the context of (15). Assume that:

(B1) $\psi_{\mathrm{s}} : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function;

(B2) $\psi_{\mathrm{n}} \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$.

Definition 2.1 *For a given $z^0 \in \mathrm{dom}\,\psi_{\mathrm{n}}$ and parameter $\sigma \in \mathbb{R}_+$, a triple $(\bar{z}, \bar{r}, \bar{\varepsilon}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+$ satisfying*

$$\bar{r} \in \nabla\psi_{\mathrm{s}}(\bar{z}) + \partial_{\bar{\varepsilon}}\psi_{\mathrm{n}}(\bar{z}) \quad \textit{and} \quad \|\bar{r}\|^2 + 2\bar{\varepsilon} \leq \sigma\|z^0 - \bar{z}\|^2 \tag{18}$$

is called a $(\sigma; z^0)$-relative stationary solution of (15) with composite term ψ_{n}.

We now make some remarks about Definition 2.1. First, Appendix C (see Proposition C.3 and the remark after it) discusses how a triple $(\bar{z}, \bar{r}, \bar{\varepsilon})$ satisfying Definition 2.1 yields a point close to $\bar{z}$ with nearly nonnegative directional derivatives along all unit directions. Second, if $\sigma = 0$, then the inequality in (18) implies that $(\bar{r}, \bar{\varepsilon}) = (0, 0)$, and hence the inclusion in (18) implies that $\bar{z}$ is an exact stationary point of (15), i.e., it satisfies $0 \in \nabla\psi_{\mathrm{s}}(\bar{z}) + \partial\psi_{\mathrm{n}}(\bar{z})$, or equivalently, the condition that the directional derivative of $\psi_{\mathrm{s}} + \psi_{\mathrm{n}}$ at $\bar{z}$ satisfies $(\psi_{\mathrm{s}} + \psi_{\mathrm{n}})'(\bar{z}; d) \geq 0$ for every $d \in \mathbb{R}^n$ (see Lemma C.2 in the Appendix). Thus, if the triple $(\bar{z}, \bar{r}, \bar{\varepsilon})$ is a $(\sigma; z^0)$-relative stationary solution of (15), then $\bar{z}$ can be viewed as an approximate stationary solution of (15) where the residual pair $(\bar{r}, \bar{\varepsilon})$ is bounded according to (18) (instead of being zero as in the exact case). Third, if $\bar{z}$ is an exact stationary point of (15) (e.g., $\bar{z}$ is an exact solution of (15)), then the triple $(\bar{z}, 0, 0)$ is a $(\sigma; z^0)$-relative stationary point of (15) for any $\sigma \in \mathbb{R}_+$.

In general, an exact solution or relative stationary point of (15) is not easy to compute. In such a case, Proposition 3.5 of [22] (see also Subsection 2.3 in [33] and Appendix A in [58]) establishes the iteration-complexity of a variant of the original Nesterov's accelerated gradient method [50] (see also [49] and [51, Chapter 2]) to find such an approximate solution under the assumptions that either $\psi_{\rm s}$ is convex or is strongly convex.

Appendix B describes the ADAP-FISTA method of [58] and its main properties (see Proposition B.1 in Appendix B). The main reason for focusing on this method instead of the ones above are: i) in addition to being able to find an approximate solution as in Definition 2.1 when $\psi_{\rm s}$ is strongly convex and its gradient is Lipschitz continuous, ADAP-FISTA is also applicable to instances where $\psi_{\rm s}$ is weakly convex (and hence possibly nonconvex), and; ii) ADAP-FISTA provides a key and easy to check inequality whose validity at every iteration guarantees its successful termination. These two properties of ADAP-FISTA play an important role in the development of adaptive ADMMs with variable prox stepsizes for solving nonconvex instances of (1) (see Section 6). Finally, ADAP-FISTA shares similar features with other accelerated gradient methods (e.g., see [15, 21, 22, 29, 33, 45, 49, 51]) in that: it has similar complexity guarantees regardless of whether it succeeds or fails (e.g., see [29, 32, 52]); it successfully terminates when $\psi_{\rm s}$ is μ-strongly convex; and it performs a line search for estimating a local Lipschitz constant for the gradient of $\psi_{\rm s}$ (e.g., see [15]).

As mentioned in Subsection 1.1, Sections 3 to 5 assume that the weak convexity parameters m_t's are known and present ADMM variants which compute their constant prox stepsize in terms of the m_t's.

3 A Fixed Penalty ADMM

This section contains two subsections. The first one describes an important component of an ADMM method, namely, a subroutine for performing a block inexact proximal point (BIPP) iteration within it, as mentioned in the paragraph containing (4) and (5). The second one presents FP-ADMM, an ADMM variant which, in addition to keeping its prox stepsize constant, also keeps its penalty parameter fixed.

3.1 The BIPP Subroutine

This subsection is to state the BIPP subroutine, its main properties, and relevant remarks about it.

We start by describing the subroutine.

Subroutine BIPP

Input: $(z, p, \lambda, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}^B_{++} \times \mathbb{R}_{++}$
Output: $(z^+, v^+, \delta_+) \in \mathcal{H} \times \mathbb{R}^l \times \mathbb{R}_{++}$

1: STEP 1: Block-IPP Iteration
2: **for** $t = 1, \ldots, B$ **do**
3: compute a $(1/8; z_t)$-relative stationary solution $(z_t^+, r_t^+, \varepsilon_t^+)$ of

$$\min_{u \in \mathbb{R}^{n_t}} \left\{ \lambda_t \hat{\mathcal{L}}_c(z^+_{<t}, u, z_{>t}; p) + \frac{1}{2}\|u - z_t\|^2 + \lambda_t h_t(u) \right\} \tag{19}$$

with composite term $\lambda_t h_t(\cdot)$ (see Definition 2.1)
4: $z^+ \leftarrow (z_1^+, \ldots, z_B^+)$

5: STEP 2: Computation of the residual pair (v^+, δ_+) for (z^+, p)
6: **for** $t = 1, \ldots, B$ **do**
7: $v_t^+ \leftarrow \nabla_t f(z^+_{<t}, z^+_t, z^+_{>t}) - \nabla_t f(z^+_{<t}, z^+_t, z_{>t}) + \dfrac{r_t^+}{\lambda_t} + cA_t^* \sum_{s=t+1}^{B} A_s(z_s^+ - z_s) - \dfrac{1}{\lambda_t}(z_t^+ - z_t)$
8: $v^+ \leftarrow (v_1^+, \ldots, v_B^+)$ and $\delta_+ \leftarrow (\varepsilon_1^+/\lambda_1) + \ldots + (\varepsilon_B^+/\lambda_B)$

9: **return** (z^+, v^+, δ_+)

We now clarify some aspects of BIPP. First, line 3 requires a subroutine to find an approximate solution of (19) as in Definition 2.1. A detailed discussion giving examples of such subroutine will be given in the second paragraph after Proposition 3.1. Second, Proposition 3.1 shows that the iterate z^+ and the residual pair (v^+, δ_+) computed in lines 4 and 8, respectively, satisfy the approximate stationary inclusion $v^+ \in \nabla f(z^+) + \partial_{\delta_+} h(z^+) + \mathrm{Im}(A^*)$. Hence, upon termination of BIPP, its output satisfies the first condition in (3) (though it may not necessarily fulfill the remaining two) and establishes an important bound on the residual pair (v^+, δ_+) in terms of a Lagrangian function variation that will be used later to determine a suitable potential function.

We now make some remarks about the prox stepsizes. First, the prox stepsizes λ are kept constant during the whole execution of BIPP. Hence, ADMMs that repeatedly invoke BIPP yield constant stepsize ADMM variants. Section 6 describes adaptive prox stepsize ADMM variants that repeatedly invoke a version of BIPP that adaptively chooses the initial stepsizes λ, and outputs λ^+ possibly different from λ.

The quantities

$$\zeta_1 := 100 \max\left\{1\,,\ \max_{1 \le t \le B} m_t\right\} + 24L^2 + 1, \qquad \zeta_2 := 24(B-1)\|A\|_\dagger^2, \tag{20}$$

where

$$L := \sqrt{\sum_{t=1}^{B-1} (L_{>t})^2}, \qquad \|A\|_\dagger := \sqrt{\sum_{t=1}^{B} \|A_t\|^2}, \tag{21}$$

with scalars $L_{>t}$'s as in (13) and submatrices A_t's as in (2), are used in the following statement of the main result of this subsection.

Proposition 3.1 *Assume that $(z^+, v^+, \delta_+) = \mathrm{BIPP}(z, p, \lambda, c)$ for some $(z, p, \lambda, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}^B_{++} \times \mathbb{R}_{++}$ and the prox stepsize $\lambda \in \mathbb{R}^B_{++}$ input to BIPP is chosen as*

$$\lambda_t = \frac{1}{2\max\{m_t, 1\}} \quad \forall t \in \{1, \ldots, B\}. \tag{22}$$

Then, the following statements hold:

- *(a) for any $t \in \{1, \ldots, B\}$, the smooth part of the objective function of the t-th block subproblem (19) is $(1/2)$-strongly convex;*
- *(b) it holds that*

$$v^+ \in \nabla f(z^+) + \partial_{\delta_+} h(z^+) + A^*[p + c(Az^+ - b)], \tag{23}$$

$$\|v^+\|^2 + \delta_+ \le (\zeta_1 + c\zeta_2)\Big[\mathcal{L}_c(z; p) - \mathcal{L}_c(z^+; p)\Big], \tag{24}$$

  where ζ_1 and ζ_2 are as in (20).

We postpone the proof of Proposition 3.1 to the end of Subsection 6.1 and, for now, only make some remarks about it. First, the inclusion (23) shows that (v^+, δ_+) is a residual pair for the point z^+. Second, the inequality in (24) provides a bound on the magnitude of the residual pair (v^+, δ_+) in terms of a variation of the Lagrangian function which, in the analysis of the next section, will play the role of a potential function.

For any given $t \in \{1, \ldots, B\}$, we now comment on the possible ways to obtain a $(1/8; z_t)$-relative stationary solution of (19) as required in line 3 of BIPP. As already observed in the paragraph following Definition 2.1, if an exact solution z_t^+ of (19) can be computed in closed form, then $(z_t^+, v_t^+, \varepsilon_t^+) = (z_t^+, 0, 0)$ is a $(1/8; z_t)$-relative stationary solution of (19). Another approach is to use ADAP-FISTA described in Appendix B. Specifically, assume that $\nabla_t f(x_1, \ldots, x_B)$ exists for every $x = (x_1, \ldots, x_B) \in \mathcal{H}$ and is $\tilde{L}_t$-Lipschitz continuous with respect to the t-th block x_t. Using this assumption and Proposition 3.1(a), it follows from statement (c) of Proposition B.1 in Appendix B with $\tilde{M} = 1 + \lambda_t(\tilde{L}_t + c\|A_t\|^2)$ that ADAP-FISTA with input $(\sigma, z^0, M_0, \mu_0) = (1/\sqrt{8}, z_t, \lambda_t c\|A_t\|^2, 1/2)$ obtains a $(1/8; z_t)$-relative stationary solution of (19). Moreover, since $M_0 \le \tilde{M}$, $\mu_0 = 1/2$, and $\lambda_t \le 1/2$ for every $t \in \{1, \ldots, B\}$, Proposition B.1(a) ensures that the number of iterations performed by ADAP-FISTA to obtain such a relative stationary solution is bounded (up to logarithmic terms) by $\mathcal{O}([\tilde{L}_t + c\|A_t\|^2]^{1/2})$. Even though this iteration-complexity bound is expressed in terms of $\tilde{L}_t$, ADAP-FISTA itself does not require $\tilde{L}_t$.

Finally, as already observed before, BIPP is a key component that is invoked once in every iteration of the non-adaptive ADMM presented in subsequent subsection. The complexity bounds for this ADMM will be given in terms of ADMM iterations (and hence BIPP calls) and will not take into account the complexities of implementing line 3. The main reason for doing so is the possible different ways of solving the block subproblems (e.g., in closed form, or using an ACG variant, or some other convex optimization solver). Nevertheless, the discussion in the previous paragraph provides ways of estimating the contribution of each block to the overall algorithmic effort.

3.2 Description of FP-ADMM

This subsection presents FP-ADMM, an ADMM variant which, in addition to keeping its penalty parameter constant, also works with a prox stepsize parameter determined by the weak convexity parameters m_t's.

FP-ADMM is shown to obtain a quadruple $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ satisfying (3) only under the condition that its (constant) penalty parameter is larger than a threshold value, which is usually unknown or difficult to estimate. Due to this difficulty and other ones discussed in the first two remarks after Theorem 3.2, a single call to FP-ADMM as a means to find a quadruple as in (3) is not recommended. Instead, the variable penalty ADMM of Section 5 is shown to obtain such a quadruple by repeatedly doubling c and warm-starting FP-ADMM. It is also shown that the iteration-complexity of VP-ADMM is better than that of FP-ADMM under the assumption that its constant penalty parameter is (somehow) suitably chosen.

FP-ADMM is formally stated next.

Algorithm 1 FP-ADMM ($m = (m_1, \ldots, m_B)$ is required)

Universal Input: $\rho > 0$, $\alpha > 0$, $C \geq \rho$, and $m = (m_1, \ldots, m_B) \in \mathbb{R}^B_+$
Input: $(y^0, q^0, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}_{++}$
Output: $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ that satisfies the conclusions of (a) and (b) of Theorem 3.2.

1: $T_0 \leftarrow 0,\ k \leftarrow 0$
2: $\lambda_t \leftarrow \dfrac{1}{2\max\{m_t, 1\}}$ for $t = 1, \ldots, B$
3: **for** $i \leftarrow 1, 2, \ldots$ **do**
4: $\quad (y^i, v^i, \delta_i) = \mathrm{BIPP}(y^{i-1}, q^{i-1}, \lambda, c)$
5: $\quad$ **if** $\|v^i\|^2 + \delta_i \leq \rho^2$ **then** ▷ termination criterion
6: $\quad\quad k \leftarrow k+1,\ \ i_k \leftarrow i$ ▷ end of the final epoch
7: $\quad\quad q^i = q^{i-1} + c(Ay^i - b)$
8: $\quad\quad$ **return** $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta}) = (y^i, q^i, v^i, \delta_i)$

9: $\quad T_i = \mathcal{L}_c(y^{i-1}; q^{i-1}) - \mathcal{L}_c(y^i; q^{i-1}) + T_{i-1}$
10: $\quad$ **if** $\|v^i\|^2 + \delta_i \leq C^2$ and $\dfrac{\rho^2}{\alpha(k+1)} \geq \dfrac{T_i}{i}$ **then**
11: $\quad\quad k \leftarrow k+1,\ \ i_k \leftarrow i$ ▷ end of epoch $\mathcal{I}_k$
12: $\quad\quad q^i = q^{i-1} + c(Ay^i - b)$ ▷ Lagrange multiplier update
13: $\quad$ **else**
14: $\quad\quad q^i = q^{i-1}$

We now explain some details about FP-ADMM. The iteration index i counts the number of iterations of FP-ADMM. Index k counts the number of Lagrange multiplier updates performed by FP-ADMM. The index i_k computed either in lines 6 or 11 of FP-ADMM is the iteration index where the k-th Lagrange multiplier occurs. It is shown in Theorem 3.2(a) that the total number of iterations performed by FP-ADMM is finite, and hence that the index i_k is well-defined. If the inequality in line 5 is satisfied, FP-ADMM performs the last Lagrange multiplier update and stops in line 8. Otherwise, depending on the test in line 10, FP-ADMM either performs a Lagrange multiplier in line 12 or leaves it unchanged in line 14, and in both cases moves on to the next iteration.

We now comment on the instance parameters required by FP-ADMM. First, it requires the weakly convex parameters $\{m_t\}_{t=1}^B$ (see its universal input), as they are used to compute the proximal stepsizes $\{\lambda_t\}_{t=1}^B$ in line 2, which remain constant throughout its execution. However, FP-ADMM requires none of the Lipschitz constant M_h as in assumption (A1), the Lipschitz constants $\{L_{>t}\}_{t=1}^{B-1}$ as in assumption (A3), or the Lipschitz constants $\{\tilde{L}_t\}_{t=1}^{B}$ mentioned in the second last paragraph of Subsection 3.1 (see the discussion therein). Subsection 6.2 presents a fully adaptive ADMM variant (i.e., one that requires none of the above parameters) which, instead of choosing the prox stepsizes in a constant manner as in line 2 of FP-ADMM, selects them adaptively.

We next define a few concepts that will be used in the discussion and analysis of FP-ADMM. For every $k \geq 1$, define the k-th epoch $\mathcal{I}_k$ as the index set

$$\mathcal{I}_k := \{i_{k-1}+1, \ldots, i_k\}, \tag{25}$$

with the convention that $i_0 = 0$. Moreover, let

$$(\tilde{y}^k, \tilde{q}^k, \tilde{\lambda}^k, \tilde{T}_k) := (y^{i_k}, q^{i_k}, \lambda^{i_k}, T_{i_k}) \quad \forall k \geq 0, \tag{26}$$

and

$$(\tilde{v}^k, \tilde{\delta}_k) := (v^{i_k}, \delta_{i_k}) \quad \forall k \geq 1. \tag{27}$$

We now make two additional remarks about the logic of FP-ADMM regarding the prox stepsize and the Lagrange multiplier. First, due to the definition of i_k, it follows that $q^i = q^{i-1}$ for every $i \in \{i_{k-1}+1, \ldots, i_k - 1\} = \mathcal{I}_k \setminus \{i_k\}$, which implies that

$$q^{i-1} = q^{i_{k-1}} = \tilde{q}^{k-1} \quad \forall i \in \mathcal{I}_k. \tag{28}$$

Moreover, (26) and (28) with $i = i_k$ imply that

$$\tilde{q}^k = q^{i_k} = q^{i_k - 1} + c(Ay^{i_k} - b) = \tilde{q}^{k-1} + c(A\tilde{y}^k - b) \quad \forall k \geq 1. \tag{29}$$

Second, $\tilde{q}^0 \in \mathrm{Im}(A)$ due to the facts that $\tilde{q}^0 = q^0$ by (26) with $k = 0$, and the input q^0 of FP-ADMM is in $\mathrm{Im}(A)$. This observation, the fact that $b \in \mathrm{Im}(A)$ by (A4), identity (29), and a simple induction argument, then imply that

$$\tilde{q}^k \in A(\mathbb{R}^n) \quad \forall k \geq 0, \tag{30}$$

and hence that

$$q^i \in A(\mathbb{R}^n) \quad \forall i \geq 0, \tag{31}$$

in view of (28).

Before stating the main result of this subsection, we define the quantities

$$\Upsilon(C) := \frac{2D_h M_h + (2D_h + 1)(C + C^2 + \widehat{\nabla}_f)}{\bar{d}\nu_A^+}, \tag{32}$$

$$\Gamma(y^0, q^0, c\,; C, \alpha) := \overline{\phi} - \underline{\phi} + c\|Ay^0 - b\|^2 + \left[\frac{4(\zeta_1 + c\zeta_2)}{\alpha} + 1\right] \frac{\|q^0\|^2 + \Upsilon^2(C)}{c}, \tag{33}$$

where (y^0, q^0, c) is the input of FP-ADMM, (ζ_1, ζ_2) is as in (20), M_h and $\bar{d}$ are as in (A1) and (A4), respectively, $(D_h, \widehat{\nabla}_f)$ is as in (14), and ν_A^+ is the smallest positive singular value of the nonzero linear operator A.

The main iteration-complexity result for FP-ADMM, whose proof is given in Section 4, is stated next.

Theorem 3.2 (FP-ADMM Complexity) *Assume that $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta}) = \text{FP-ADMM}(y^0, q^0, c)$ for some triple $(y^0, q^0, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}_{++}$. Then, for any tolerance pair $(\rho, \eta) \in \mathbb{R}^2_{++}$ and $C \geq \rho$, the following statements hold for* FP-ADMM*:*

(a) its total number of iterations (and hence BIPP *calls) is bounded by*

$$\left(\frac{\zeta_1 + c\zeta_2}{\rho^2}\right) \Gamma(y^0, q^0, c\,; C, \alpha) + 1, \tag{34}$$

where (ζ_1, ζ_2) and $\Gamma(y^0, q^0, c\,; C, \alpha)$ are as in (20) and (33), respectively;

(b) its output $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ satisfies

$$\hat{v} \in \nabla f(\hat{y}) + \partial_{\hat{\delta}} h(\hat{y}) + A^* \hat{q} \quad \text{and} \quad \|\hat{v}\|^2 + \hat{\delta} \le \rho^2, \tag{35}$$

and the following bounds

$$c\|A\hat{y} - b\| \le 2\max\{\|q^0\|, \Upsilon(C)\} \quad \text{and} \qquad \|\hat{q}\| \le \max\{\|q^0\|, \Upsilon(C)\}, \tag{36}$$

where $\Upsilon(C)$ is as in (32);

(c) if $c \ge 2\max\{\|q^0\|, \Upsilon(C)\}/\eta$, then the output $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ of FP-ADMM *is a (ρ,η)-stationary solution of problem (1)-(2) according to (3).*

We now make some remarks about Theorem 3.2. First, Theorem 3.2(b) implies that FP-ADMM returns a quadruple $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ satisfying both conditions in (35), but not necessarily the feasibility condition $\|A\hat{y}-b\| \le \eta$. However, Theorem 3.2(c) guarantees that, if c is chosen large enough, i.e., $c = \Omega(\eta^{-1})$, then the feasibility also holds, and hence that $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta})$ is a (ρ,η)-stationary solution of (1)-(2).

Second, if $\alpha = \Omega(1)$ and $C = \mathcal{O}(1)$ then it follows from (33) that $\Gamma(y^0, q^0, c\,; C, \alpha) = \mathcal{O}(1 + c\|Ay^0 - b\|^2)$, and hence (34) implies that the overall complexity of FP-ADMM is

$$\mathcal{O}\left(\frac{1+c}{\rho^2}\left(1 + c\|Ay^0 - b\|^2\right)\right).$$

If the initial point y^0 satisfies $c\|Ay^0 - b\|^2 = \mathcal{O}(1)$, then the above bound reduces to $\mathcal{O}((1+c)\rho^{-2})$. Moreover, under the assumption made in Theorem 3.2(c), i.e., that $c = \Theta(\eta^{-1})$, then the above two complexity estimates reduce to $\mathcal{O}(\eta^{-2}\rho^{-2})$ if y^0 satisfies $\|Ay^0 - b\| = \mathcal{O}(1)$ and to $\mathcal{O}(\eta^{-1}\rho^{-2})$ if y^0 satisfies $c\|Ay^0 - b\|^2 = \mathcal{O}(1)$.

Third, it is worth discussing the dependence of the complexity bound (34) in terms of number of blocks B only. Observe that the only quantity in (33) that depends on B is the quantity ζ_2 defined in (20), and hence satisfies $\zeta_2 = \Theta(B)$. Thus, $\Gamma(y^0, q^0, c\,; C, \alpha) = \mathcal{O}(1 + B/\alpha)$ due to (33), and hence the complexity bound (34) is $\mathcal{O}(B(1 + B\alpha^{-1}))$. So, if α is chosen to be $\alpha = \Omega(B)$, then the dependence of (34), in terms of B only, is $\mathcal{O}(B)$.

Section 5 presents an ADMM variant, namely VP-ADMM, which gradually increases the penalty parameter and achieves the complexity bound $\mathcal{O}(\eta^{-1}\rho^{-2})$ of the previous paragraph for any y^0 such that $\|Ay^0 - b\| = \mathcal{O}(1)$. Specifically, VP-ADMM repeatedly doubles c and warm-starts FP-ADMM, i.e., if c is the penalty parameter used in the previous FP-ADMM call and $(\hat{x}, \hat{p}, \hat{v}, \hat{\varepsilon})$ is its output, then the current FP-ADMM call uses $(\hat{x}, \hat{p}, 2c)$ as input.

4 The Proof of FP-ADMM's Complexity (Theorem 3.2)

The first result of this section shows that every iterate $(y^i, v^i, \delta_i, \lambda^i)$ of FP-ADMM satisfies the stationary inclusion $v^i \in \nabla f(y^i) + \partial_{\delta_i} h(y^i) + \mathrm{Im}(A^*)$ and derives a bound on the residual error (v^i, δ_i).

Lemma 4.1 *Consider the sequence $\{(y^i, q^i, v^i, \delta_i, T_i)\}$ generated by* FP-ADMM*. Then, for every iteration index $i \ge 1$, we have*

$$v^i \in \nabla f(y^i) + \partial_{\delta_i} h(y^i) + A^*[q^{i-1} + c(Ay^i - b)] \tag{37}$$

$$\frac{1}{\zeta_1 + c\zeta_2}(\|v^i\|^2 + \delta_i) \le \mathcal{L}_c(y^{i-1}; q^{i-1}) - \mathcal{L}_c(y^i; q^{i-1}) = T_i - T_{i-1}, \tag{38}$$

where (ζ_1, ζ_2) is as in (20).

Proof: The proofs of both (37) and (38) are based on Proposition 3.1 with $(z, p, \lambda, c) = (y^{i-1}, q^{i-1}, \lambda, c)$ and the fact that $(y^i, v^i, \delta_i) = \mathrm{BIPP}(y^{i-1}, q^{i-1}, \lambda, c)$ due to line 4 of FP-ADMM. Specifically, (37) follows from the conclusion (23), and (38) follows from (24) and the fact that λ is chosen as in line 2 of FP-ADMM. ∎

We now make some remarks about Lemma 4.1. First, (38) implies that: $\{T_i\}$ is nondecreasing; and, if $T_i = T_{i-1}$, then $(v^i, \delta_i) = (\mathbf{0}, 0)$, which together with (37), implies that the algorithm stops in line 8 with an exact stationary point for problem (1)-(2). In view of this remark, it is natural to view $\{T_i\}$ as a potential sequence. Second, if $\{T_i\}$ is bounded, (38) immediately implies that the quantity $\|v^i\|^2 + \delta_i$ converges to zero, and hence that y^i eventually becomes a near stationary point for problem (1)-(2), again in view of (37). A major effort of our analysis below will be to show that $\{T_i\}$ is bounded.

Among the results of this section, as well as the ones in Section 5, only Lemma 4.1 uses the fact that λ is chosen according to (22). All other results only require that the following weaker condition hold:

Condition C1: there exists $(\zeta_1, \zeta_2) \in \mathbb{R}^2_{++}$ such that (37) and (38) hold for every $i \geq 1$.

Moreover, the fact that the pair (ζ_1, ζ_2) is chosen as in (20) plays no role in the proofs of these results. Thus, if condition **C1** holds for some pair $(\zeta_1, \zeta_2) \in \mathbb{R}^2_{++}$ other than the one given by (20), then the conclusions of Theorems 3.2 and 5.2 still hold for this pair. These observations will be used in Section 6 to establish the complexity of an ADMM variant that chooses the prox stepsizes adaptively, and hence not according to line 2 of FP-ADMM.

The first result gives an expression for T_i that plays an important role in our analysis.

Lemma 4.2 *If i is an iteration index generated by* FP-ADMM *such that $i \in \mathcal{I}_k$, then*

$$T_i = \left[\mathcal{L}_c(\tilde{y}^0; \tilde{q}^0) - \mathcal{L}_c(y^i; \tilde{q}^{k-1})\right] + \frac{1}{c}\sum_{\ell=1}^{k-1} \|\tilde{q}^\ell - \tilde{q}^{\ell-1}\|^2. \tag{39}$$

Proof: We first note that

$$\begin{aligned} T_i - T_1 &= \sum_{j=2}^{i}(T_j - T_{j-1}) = \sum_{j=1}^{i-1}(T_{j+1} - T_j) = \sum_{j=1}^{i-1}\left[\mathcal{L}_c(y^j; q^j) - \mathcal{L}_c(y^{j+1}; q^j)\right] \\ &= \sum_{j=1}^{i-1}\left[\mathcal{L}_c(y^j; q^j) - \mathcal{L}_c(y^j; q^{j-1})\right] + \sum_{j=1}^{i-1}\left[\mathcal{L}_c(y^j; q^{j-1}) - \mathcal{L}_c(y^{j+1}; q^j)\right]. \end{aligned} \tag{40}$$

Moreover, using the definition of T_i with $i = 1$ (see line 9 of FP-ADMM), the fact that $q^{i-1} = \tilde{q}^{k-1}$ due to (28) and simple algebra, we have

$$\begin{aligned} &T_1 + \sum_{j=1}^{i-1}\left[\mathcal{L}_c(y^j; q^{j-1}) - \mathcal{L}_c(y^{j+1}; q^j)\right] = T_1 + \mathcal{L}_c(y^1; q^0) - \mathcal{L}_c(y^i; q^{i-1}) \\ &= [\mathcal{L}_c(y^0; q^0) - \mathcal{L}_c(y^1; q^0)] + [\mathcal{L}_c(y^1; q^0) - \mathcal{L}_c(y^i; \tilde{q}^{k-1})] = \mathcal{L}_c(y^0; q^0) - \mathcal{L}_c(y^i; \tilde{q}^{k-1}). \end{aligned} \tag{41}$$

Using the definition of the Lagrangian function (see definition in (4)), relations (29) and (28), we conclude that for any $\ell \leq k$,

$$\mathcal{L}_c(y^j; q^j) - \mathcal{L}_c(y^j; q^{j-1}) \overset{(4)}{=} \left\langle Ay^j - b\,, q^j - q^{j-1}\right\rangle \overset{(29),(28)}{=} \begin{cases} 0 & \text{, if } j \in \mathcal{I}_\ell \setminus \{i_\ell\}; \\ \dfrac{\|\tilde{q}^\ell - \tilde{q}^{\ell-1}\|^2}{c} & \text{, if } j = i_\ell, \end{cases}$$

and hence that

$$\sum_{j=1}^{i-1}\left[\mathcal{L}_c(y^j; q^j) - \mathcal{L}_c(y^j; q^{j-1})\right] = \frac{1}{c}\sum_{\ell=1}^{k-1}\|\tilde{q}^\ell - \tilde{q}^{\ell-1}\|^2.$$

Identity (39) now follows by combining the above identity with the ones in (40) and (41). ■

The next technical result will be used to establish an upper bound on the first term of the right-hand side of (39).

Lemma 4.3 *For any given $c > 0$ and pairs $(u, p) \in \mathcal{H} \times \mathbb{R}^l$ and $(\tilde{u}, \tilde{p}) \in \mathcal{H} \times \mathbb{R}^l$, we have*

$$\mathcal{L}_c(u; p) - \mathcal{L}_c(\tilde{u}; \tilde{p}) \leq \overline{\phi} - \underline{\phi} + c\|Au - b\|^2 + \frac{1}{2c}\max\{\|p\|, \|\tilde{p}\|\}^2 \tag{42}$$

where $(\overline{\phi}, \underline{\phi})$ is as in (14).

Proof: Using the definitions of $\mathcal{L}_c(\cdot\,;\,\cdot)$ and $\underline{\phi}$ as in (4) and (14), respectively, we have

$$\begin{aligned}\mathcal{L}_c(\tilde{u};\tilde{p}) - \underline{\phi} &\stackrel{(14)}{\geq} \mathcal{L}_c(\tilde{u};\tilde{p}) - (f+h)(\tilde{u}) \\ &\stackrel{(4)}{=} \langle \tilde{p}, A\tilde{u} - b\rangle + \frac{c}{2}\|A\tilde{u}-b\|^2 = \frac{1}{2}\left\|\frac{\tilde{p}}{\sqrt{c}} + \sqrt{c}(A\tilde{u}-b)\right\|^2 - \frac{\|\tilde{p}\|^2}{2c} \geq -\frac{\|\tilde{p}\|^2}{2c}.\end{aligned}$$

On the other hand, using the definitions of $\mathcal{L}_c(\,\cdot\,;\,\cdot\,)$ and $\overline{\phi}$ as in (4) and (14), respectively, and the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\mathcal{L}_c(u;p) - \overline{\phi} &\stackrel{(14)}{\leq} \mathcal{L}_c(u;p) - (f+h)(u) \stackrel{(4)}{=} \langle p, Au-b\rangle + \frac{c\|Au-b\|^2}{2} \\ &\leq \left(\frac{\|p\|^2}{2c} + \frac{c\|Au-b\|^2}{2}\right) + \frac{c\|Au-b\|^2}{2} = \frac{\|p\|^2}{2c} + c\|Au-b\|^2.\end{aligned}$$

Combining the above two relations, we then conclude that (42) holds. ■

The following result shows that the number of epochs, the number of iterations, and the sequence $\{T_i\}$, generated by FP-ADMM are all suitably bounded.

Proposition 4.4 *The following statements about* FP-ADMM *hold:*

(a) its total number E *of epochs is bounded by* $\lceil(\zeta_1 + c\zeta_2)/\alpha\rceil$ *where* ζ_1 *and* ζ_2 *are as in* (20)*;*

(b) for every iteration index i*, we have* $T_i \leq \Lambda_{\mathrm{E}}(y^0;c)$*;*

(c) the number of iterations performed by FP-ADMM *is bounded by*

$$1 + \left(\frac{\zeta_1 + c\zeta_2}{\rho^2}\right)\Lambda_{\mathrm{E}}(y^0;c), \tag{43}$$

where

$$\Lambda_{\mathrm{E}}(y^0;c) := \overline{\phi} - \underline{\phi} + c\|Ay^0 - b\|^2 + \frac{Q_{\mathrm{E}}^2}{2c} + \frac{(\zeta_1 + c\zeta_2)F_{\mathrm{E}}^2}{c\alpha}, \tag{44}$$

and

$$\begin{aligned} Q_{\mathrm{E}} &:= \max\left\{\|\tilde{q}^k\| : k \in \{0,\ldots,\mathrm{E}-1\}\right\}, \\ F_{\mathrm{E}} &:= \begin{cases} 0 & , \text{ if } \mathrm{E} = 1 \\ \max\left\{\|\tilde{q}^k - \tilde{q}^{k-1}\| : k \in \{1,\ldots,\mathrm{E}-1\}\right\} & , \text{ if } \mathrm{E} \neq 1. \end{cases} \end{aligned} \tag{45}$$

Proof: (a) Assume for the sake of contradiction that FP-ADMM generates an epoch $\mathcal{I}_K$ such that $K > \lceil(\zeta_1 + c\zeta_2)/\alpha\rceil$, and hence $K \geq 2$. Using the fact that i_{K-1} is the last index of $\mathcal{I}_{K-1}$ and noting the epoch termination criteria in line 10 of FP-ADMM, we then conclude that $\tilde{T}_{K-1}/i_{K-1} \leq \rho^2/K$. Also, since FP-ADMM did not terminate during epochs $\mathcal{I}_1,\ldots,\mathcal{I}_{K-1}$, it follows from its termination criterion in line 5 that $\|v^i\|^2 + \delta_i > \rho^2$ for every iteration $i \leq i_{K-1}$. These two previous observations, (38) with $i \in \{1,\ldots,i_{K-1}\}$, the facts that $T_0 = 0$ by definition and $T_{i_{K-1}} = \tilde{T}_{K-1}$ due to (26), imply that

$$\rho^2 < \frac{1}{i_{K-1}}\sum_{i=1}^{i_{K-1}}(\|v^i\|^2 + \delta_i) \stackrel{(38)}{\leq} \frac{\zeta_1 + c\zeta_2}{i_{K-1}}\sum_{i=1}^{i_{K-1}}(T_i - T_{i-1}) = \frac{(\zeta_1 + c\zeta_2)\tilde{T}_{K-1}}{i_{K-1}} \leq \frac{(\zeta_1 + c\zeta_2)}{\alpha K}\rho^2.$$

Since this inequality and the assumption (for the contradiction) that $K > \lceil(\zeta_1 + c\zeta_2)/\alpha\rceil$ yield an immediate contradiction, the conclusion of the statement follows.

(b) Since $\{T_i\}$ is nondecreasing, it suffices to show that $T_i \leq \Lambda_{\mathrm{E}}(y^0;c)$ holds for any $i \in \mathcal{I}_{\mathrm{E}}$, where E is the total number of epochs of FP-ADMM (see statement (a)). It follows from the definition of Q_{E}, and Lemma 4.3 with $(u,p) = (\tilde{y}^0;\tilde{q}^0) = (y^0,q^0)$ (due to (26)) and $(\tilde{u},\tilde{p}) = (y^i;\tilde{q}^{\mathrm{E}-1})$, that

$$\mathcal{L}_c(\tilde{y}^0;\tilde{q}^0) - \mathcal{L}_c(y^i;\tilde{q}^{\mathrm{E}-1}) \leq \overline{\phi} - \underline{\phi} + c\|Ay^0 - b\|^2 + \frac{1}{2c}Q_{\mathrm{E}}^2.$$

Now, using the definition of $F_{\rm E}$ as in (45), we have that

$$\frac{1}{c}\sum_{k=1}^{{\rm E}-1}\|\tilde q^k-\tilde q^{k-1}\|^2\le\frac{({\rm E}-1)}{c}F_{\rm E}^2\le\frac{(\zeta_1+c\zeta_2)}{c\alpha}F_{\rm E}^2,$$

where the last inequality follows from the fact that ${\rm E}-1\le(\zeta_1+c\zeta_2)/\alpha$ due to statement (a). The inequality $T_i\le\Lambda_{\rm E}(y^0;c)$ now follows from the two inequalities above and identity (39) with $k={\rm E}$.

(c) Assume by contradiction that there exists an iteration index i generated by FP-ADMM such that

$$i>\left(\frac{\zeta_1+c\zeta_2}{\rho^2}\right)\Lambda_{\rm E}(y^0;c)+1. \tag{46}$$

Since FP-ADMM does not stop at any iteration index smaller than i, the stopping criterion in line 5 is violated at these iterations, i.e., $\|v^j\|^2+\delta_j>\rho^2$ for every $j\le i-1$. Hence, it follows from (38), the previous inequality, the fact that $T_0=0$ due to line 1 of FP-ADMM, and statement (b) that

$$\frac{(i-1)\rho^2}{\zeta_1+c\zeta_2}<\frac{1}{\zeta_1+c\zeta_2}\sum_{j=1}^{i-1}(\|v^j\|^2+\delta_j)\overset{(38)}{\le}\sum_{j=1}^{i-1}(T_j-T_{j-1})=T_{i-1}-T_0\le T_i\le\Lambda_{\rm E}(y^0,c),$$

which contradicts (46). Thus, statement (c) holds. ∎

We now make some remarks about Proposition 4.4. First, Proposition 4.4(a) shows that the number of epochs depends linearly on c. Second, Proposition 4.4(c) shows that the total number of iterations of FP-ADMM is bounded but the derived bound is given in terms of the quantities $Q_{\rm E}$ and $F_{\rm E}$ in (45), both of which depend on the magnitude of the sequence of generated Lagrange multipliers $\{\tilde q_k : k=1,\ldots,E\}$. Hence, the bound in (43) is algorithm-dependent in that it depends on the sequence $\{\tilde q^k\}$ generated by FP-ADMM.

In what follows, our goal is to derive a bound on the total number of iterations performed by FP-ADMM that depends only on the instance of (1)–(2). With this goal in mind, we first establish two technical results related to the boundedness of the sequence of Lagrange multipliers generated by FP-ADMM.

Lemma 4.5 *Consider the following subset of epoch $\mathcal{I}_k$, for some $k\ge1$, defined as*

$$\mathcal{I}_k(C):=\{i\in\mathcal{I}_k:\|v^i\|^2+\delta_i\le C^2\}, \tag{47}$$

where $C>0$ is part of the input for FP-ADMM *and (v^i,δ_i) is as in line 8 of* BIPP. *Then, the pair $(\tilde q^{k-1},y^i)$ satisfies*

$$\|\tilde q^{k-1}+cA(y^i-b)\|\le\max\{\|\tilde q^{k-1}\|,\Upsilon(C)\},\quad\forall i\in\mathcal{I}_k(C), \tag{48}$$

where $\Upsilon(C)$ is as in (32). As a consequence,

$$\|\tilde q^k\|\le\max\{\|\tilde q^{k-1}\|,\Upsilon(C)\}. \tag{49}$$

Proof: We first argue that (49) follows as an immediate consequence of (48). Indeed, noting that the assumption that $C\ge\rho$ (see the input of FP-ADMM) and the logic of FP-ADMM imply that i_k always satisfies $\|v^{i_k}\|^2+\delta_{i_k}\le C^2$, i.e., $i_k\in\mathcal{I}_k(C)$, and using (48) with $i=i_k$, and identities (26) and (29), we immediately see that (49) holds.

To show (48), let $i\in\mathcal{I}_k(C)$ be given. The proof of (48) relies on Lemma A.3 with $(q^-,\varrho)=(\tilde q^{k-1},c)$ and $(z,q,r,\delta)=(y^i,\underline{q}^i,r^i,\delta_i)$, where

$$\underline{q}^i:=\tilde q^{k-1}+c(Ay^i-b)\quad\text{and}\quad r^i:=v^i-\nabla f(y^i). \tag{50}$$

We first claim that $(q^-,\varrho)=(\tilde q^{k-1},c)$ and $(z,q,r,\delta)=(y^i,\underline{q}^i,r^i,\delta_i)$ satisfy the assumptions of Lemma A.3, i.e., the inclusion $\tilde q^{k-1}\in A(\mathbb{R}^n)$ and the conditions in (76). Indeed, the inclusion is due to (30). Moreover, the identity in (76) is due to the first identity in (50). Also, Lemma 4.1(a), and relations (28) and (50), imply that $r^i\in\partial_{\delta_i}h(y^i)+A^*\underline{q}^i$. Thus, the claim holds.

Now, noting that the triangle inequality for norms, the second identity in (50), the definitions of $\mathcal{I}_k(C)$ in (47) and $\widehat\nabla_f$ in (14), imply that

$$\delta_i+\|r^i\|\overset{(50)}{\le}\delta_i+\|v^i\|+\|\nabla f(y^i)\|\overset{(47)}{\le}C^2+(C+\|\nabla f(y^i)\|)\overset{(14)}{\le}C^2+(C+\widehat\nabla_f), \tag{51}$$

and using the conclusion of Lemma A.3, we conclude that

$$\|\underline{q}^i\| \overset{(77)}{\leq} \max\left\{\|\tilde{q}^{k-1}\|\,,\, \Xi(\|y^i - \bar{\mathrm{x}}\|\,,\, \|r^i\| + \delta_i)\right\}$$
$$\overset{(51)}{\leq} \max\left\{\|\tilde{q}^{k-1}\|\,,\, \Xi(D_h\,,\, C + C^2 + \widehat{\nabla}_f)\right\} \leq \max\{\|\tilde{q}^{k-1}\|\,,\, \Upsilon(C)\},$$

where the second inequality is due to (51), the definition of D_h in (14), and the fact that the function $\Xi(\cdot,\cdot)$ defined in (78) is non-decreasing, and the last inequality is due to the definition of $\Xi(\cdot\,;\cdot)$ and the definition of $\Upsilon(\cdot)$ in (32). ∎

The proof of Lemma 4.5 clearly uses the assumption that $\operatorname{dom} h$ is bounded (see A1). However, the above proof shows that the conclusion of Lemma 4.5 also holds under the weaker assumption that the sequence $\{y^i\}$ generated by FP-ADMM is bounded, and D_h and $\widehat{\nabla}_f$ are instead defined as $D_h := \sup_i \|y^i - \bar{\mathrm{x}}\|$ and $\widehat{\nabla}_f := \sup_i \|\nabla f(y^i)\|$, respectively.

Lemma 4.6 *For every epoch k generated by* FP-ADMM*, we have*

$$\|\tilde{q}^k\| \leq \max\{\|q^0\|, \Upsilon(C)\}, \tag{52}$$
$$c\|Ay^i - b\| \leq 2\max\{\|q^0\|, \Upsilon(C)\}, \quad \forall i \in \mathcal{I}_k(C), \tag{53}$$
$$\|\tilde{q}^k - \tilde{q}^{k-1}\| \leq 2\max\{\|q^0\|, \Upsilon(C)\}, \tag{54}$$

where $\Upsilon(C)$ and $\mathcal{I}_k(C)$ are as in (32) *and* (47)*, respectively.*

Proof: Inequality (52) follows by recursively using (49) and the fact that $\tilde{q}^0 = q^0$ due to (26). Inequality (53) follows from (48), the triangle inequality, (52) with $k = k-1$, and the fact that $\tilde{q}^0 = q^0$ due to (26), i.e.,

$$c\|Ay^i - b\| \overset{(48)}{\leq} \max\{\|\tilde{q}^{k-1}\|, \Upsilon(C)\} + \|\tilde{q}^{k-1}\| \overset{(52)}{\leq} 2\max\{\|\tilde{q}^0\|, \Upsilon(C)\}.$$

Inequality (54) follows from identity (29), the triangle inequality for norms, (53) with $i = i_k$, and the fact that $\tilde{y}^k = y^{i_k}$ due to (26). ∎

Proof of Theorem 3.2: (a) It follows from Proposition 4.4(c) that the total number of iterations generated by FP-ADMM is bounded by the expression in (43). Now, recalling that E is the last epoch generated by FP-ADMM, and using (45), (52) and (54), we conclude that $Q_{\mathrm{E}} \leq \max\{\|q^0\|, \Upsilon(C)\}$ and $F_{\mathrm{E}} \leq 2\max\{\|q^0\|, \Upsilon(C)\}$, and hence that $\Lambda_{\mathrm{E}}(y^0; c) \leq \Gamma(y^0, q^0, c\,; C, \alpha)$, where $\Lambda_{\mathrm{E}}(y^0; c)$ and $\Gamma(y^0, q^0, c\,; C, \alpha)$ are as in (44) and (33), respectively. The conclusion now follows from the two previous observations.

(b) We first prove that the inclusion in (35) holds. It follows from (37) with $i = i_{\mathrm{E}}$, (26) and (27) with $k = \mathrm{E}$ that

$$\tilde{v}^{\mathrm{E}} \in \nabla f(\tilde{y}^{\mathrm{E}}) + \partial_{\tilde{\delta}_{\mathrm{E}}} h(\tilde{y}^{\mathrm{E}}) + A^*[q^{i_{\mathrm{E}}-1} + c(A\tilde{y}^{\mathrm{E}} - b)].$$

Using (28) with $i = i_{\mathrm{E}}$, (29) with $k = \mathrm{E}$, and the fact that $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta}) = (\tilde{y}^{\mathrm{E}}, \tilde{q}^{\mathrm{E}}, \tilde{v}^{\mathrm{E}}, \tilde{\delta}_{\mathrm{E}})$, we conclude that the inclusion in (35) holds. The inequality in (35) follows from the fact that FP-ADMM terminates in line 5 with the condition $\|\hat{v}\|^2 + \hat{\delta} = \|v^{i_{\mathrm{E}}}\|^2 + \delta_{i_{\mathrm{E}}} \leq \rho^2$ satisfied. The first inequality in (36) follows from (53) with $i = i_{\mathrm{E}}$ and the fact that $\tilde{y}^{\mathrm{E}} = y^{i_{\mathrm{E}}}$ due to (26). Finally, the second inequality in (36) follows from (52) and the fact that $(y^{i_{\mathrm{E}}}, q^{i_{\mathrm{E}}}) = (\tilde{y}^{\mathrm{E}}, \tilde{q}^{\mathrm{E}})$ due to (26).

(c) Using the assumption that $c \geq 2\max\{\|q^0\|, \Upsilon(C)\}/\eta$, statement (b) guarantees that FP-ADMM outputs $\hat{y} = y^{i_k}$ satisfying $\|A\hat{y} - b\| \leq [2\max\{\|q^0\|, \Upsilon(C)\}]/c \leq \eta$. Hence, the conclusion that $(\hat{y}, \hat{q}, \hat{v}, \hat{\delta}) = (\tilde{y}^k, \tilde{q}^k, \tilde{v}^k, \tilde{\delta}_k)$ satisfies (3) follows from the previous inequality, the inclusion in (35), and the last inequality in (36). ∎

5 A Variable Penalty ADMM

This section describes a variable penalty ADMM, or VP-ADMM for short, and establishes its iteration-complexity. In contrast to FP-ADMM, which keeps the penalty parameter constant, VP-ADMM adaptively changes the penalty parameter. The version of VP-ADMM presented in this section keeps the prox stepsizes constant throughout since it performs multiple calls to the FP-ADMM which, as already observed, also has this same attribute. An adaptive variant of VP-ADMM with variable prox stepsizes is presented in Section 6.

VP-ADMM is formally stated next.

Algorithm 2 VP-ADMM ($m = (m_1, \ldots, m_B)$ is required)

Universal Input: $(\rho, \eta) \in \mathbb{R}^2_{++}$, $\alpha > 0$, $C \geq \rho$, and $m = (m_1, \ldots, m_B) \in \mathbb{R}^B_+$
Input: $x^0 \in \mathcal{H}$
Output: $(\hat{x}, \hat{p}, \hat{u}, \hat{\varepsilon}, \hat{c})$ that satisfies the conclusion of Theorem 5.2.

1: $p^0 = (p^0_1, \ldots, p^0_B) \leftarrow (0, \ldots, 0)$ and $c_0 \leftarrow 1/[1 + \|Ax^0 - b\|]$
2: **for** $\ell \leftarrow 1, 2, \ldots$ **do**
3: $\quad (x^\ell, p^\ell, u^\ell, \varepsilon_\ell) = \text{FP-ADMM}(x^{\ell-1}, p^{\ell-1}, c_{\ell-1})$
4: $\quad c_\ell = 2c_{\ell-1}$
5: $\quad$ **if** $\|Ax^\ell - b\| \leq \eta$ **then**
6: $\quad\quad$ **return** $(\hat{x}, \hat{p}, \hat{u}, \hat{\varepsilon}, \hat{c}) = (x^\ell, p^\ell, u^\ell, \varepsilon_\ell, c_\ell)$

We now make some remarks about VP-ADMM. First, similar to Algorithm 1, VP-ADMM requires the weakly convex parameters $\{m_t\}_{t=1}^B$ to be known (see its universal input). Even though these parameters do not appear in the main body of VP-ADMM, they are used within every call to FP-ADMM (see line 3) to compute the prox stepsize λ (see line 2 of FP-ADMM). Second, similar to Algorithm 1, VP-ADMM requires none of the Lipschitz constant M_h as in assumption (A1), the Lipschitz constants $\{L_{>t}\}_{t=1}^{B-1}$ as in assumption (A3), or the Lipschitz constants $\{\tilde{L}_t\}_{t=1}^B$ mentioned in the second last paragraph of Subsection 3.1. Third, even though an initial penalty parameter c_0 is prescribed in line 1 for the sake of analysis simplification, VP-ADMM can be equally shown to converge for other choices of c_0. Fourth, it uses a "warm-start" strategy for calling FP-ADMM, i.e., after the first call to FP-ADMM, the input of any FP-ADMM call is the output of the previous FP-ADMM call.

Lemma 5.1 below and Theorem 3.2(b) imply that each FP-ADMM call in line 3 of VP-ADMM generates a quadruple $(x^\ell, p^\ell, u^\ell, \varepsilon_\ell)$ satisfying the first two conditions in (3), but not necessarily the last one, i.e., the feasibility condition which is tested in line 5. To ensure that this condition is also attained, VP-ADMM doubles the penalty parameter c_ℓ (see its line 4) every iteration. Since the first inequality in (36) ensures that $\|Ax^\ell - b\| = \mathcal{O}(1/c_\ell)$, this penalty update scheme guarantees that the test in line 5 will eventually be satisfied, and VP-ADMM will terminate with a (ρ, η)-stationary solution of (1)-(2).

Before describing the main result, we define the following constant, which appears in the total iteration-complexity,

$$\bar{\Gamma}(x^0; C, \alpha) := \overline{\phi} - \underline{\phi} + \frac{8\zeta_2 \Upsilon^2(C)}{\alpha} + 2\Upsilon^2(C)\left(\frac{4\zeta_1}{\alpha} + 9\right)(1 + \|Ax^0 - b\|), \tag{55}$$

where (ζ_1, ζ_2) is as in (20) and $\Upsilon(C)$ is as in (32).

Recalling that every VP-ADMM iteration makes an FP-ADMM call, the following result translates the properties of FP-ADMM established in Theorem 3.2 to the context of VP-ADMM.

Lemma 5.1 *Let ℓ be an iteration index of* VP-ADMM*. Then, the following statements hold:*

(a) the sequences $\{(x^k, p^k, u^k, \varepsilon_k)\}_{k=1}^\ell$ and $\{c_k\}_{k=1}^\ell$ satisfy

$$u^k \in \nabla f(x^k) + \partial_{\varepsilon_k} h(x^k) + A^* p^k \quad \textit{and} \quad \max_{1 \leq k \leq \ell} \|u^k\|^2 + \varepsilon_k \leq \rho^2, \tag{56}$$

and the following bounds

$$\max_{1 \leq k \leq \ell} \|p^k\| \leq \Upsilon(C) \quad \textit{and} \quad \max_{1 \leq k \leq \ell} c_k \|Ax^k - b\| \leq 4\Upsilon(C); \tag{57}$$

(b) the number of iterations performed by the FP-ADMM *call within the ℓ-th iteration of* VP-ADMM *(see line 3 of* VP-ADMM*) is bounded by*

$$\left(\frac{\zeta_1 + c_{\ell-1}\zeta_2}{\rho^2}\right)\bar{\Gamma}(x^0; C, \alpha) + 1, \tag{58}$$

where $\bar{\Gamma}(x^0; C, \alpha)$ is as in (55);

(c) *if $c_\ell \geq 4\Upsilon(C)/\eta$ then $(x^\ell, p^\ell, u^\ell, \varepsilon_\ell)$ is a (ρ,η)-stationary solution of problem* (1)-(2).

Proof: (a) Using Theorem 3.2(b) with $(y^0, q^0, c) = (x^{k-1}, p^{k-1}, c_{k-1})$ and noting line 3 of VP-ADMM, we conclude that for any $k \in \{1, \ldots, \ell\}$, the quadruple $(x^k, p^k, u^k, \varepsilon_k)$ satisfies (56) and the conditions

$$\|p^k\| \leq \max\{\|p^{k-1}\|, \Upsilon(C)\}, \quad c_{k-1}\|Ax^k - b\| \leq 2\max\{\|p^{k-1}\|, \Upsilon(C)\}. \tag{59}$$

A simple induction argument applied to the first inequality in (59), with the fact that $p^0 = 0$, show that the first inequality in (57) holds. The second inequality in (59), the assumption that $p^0 = 0$, the fact that $c_k = 2c_{k-1}$ for every $k \in \{1, \ldots, \ell\}$, and the first inequality in (57), imply that the second inequality in (57) also holds.

(b) Theorem 3.2(a) with $(y^0, q^0, c) = (x^{\ell-1}, p^{\ell-1}, c_{\ell-1})$ implies that the total number of iterations performed by the FP-ADMM call within the ℓ-th iteration of VP-ADMM is bounded by

$$\left(\frac{\zeta_1 + c_{\ell-1}\zeta_2}{\rho^2}\right)\Gamma(x^{\ell-1}, p^{\ell-1}, c_{\ell-1}; C, \alpha) + 1,$$

where $\Gamma(\cdot,\cdot,\cdot; C, \alpha)$ is as in (33). Thus, to show (58), it suffices to show that $\Gamma(x^{\ell-1}, p^{\ell-1}; c_{\ell-1}; C, \alpha) \leq \bar\Gamma(x_0; C, \alpha)$.

Before showing the above inequality, we first show that $c_{\ell-1}\|Ax^{\ell-1} - b\|^2 \leq 16\Upsilon^2(C)/c_0$ for every index ℓ. Indeed, this observation trivially holds for $\ell = 1$ due to the fact that $c_0 = 1/(1 + \|Ax^0 - b\|) \leq 1$ (see line 1 of VP-ADMM) and the assumption that $\Upsilon(C) \geq 1$. Moreover, the second inequality in (57) and the fact that $c_{\ell-1} \geq c_0$ show that the inequality also holds for $\ell > 1$, and thus it holds for any $\ell \geq 1$.

Using the last conclusion, the definition of $\Gamma(x^{\ell-1}, p^{\ell-1}, c_{\ell-1}; C, \alpha)$, the fact that $c_{\ell-1} \geq c_0$, and the first inequality in (57), we have

$$\begin{aligned}
\Gamma(x^{\ell-1}, p^{\ell-1}, c_{\ell-1}; C, \alpha) &\leq \overline{\phi} - \underline{\phi} + \frac{16\Upsilon^2(C)}{c_0} + \left[\frac{4\zeta_1}{\alpha c_0} + \frac{1}{c_0} + \frac{4\zeta_2}{\alpha}\right]\left(\|p^{\ell-1}\|^2 + \Upsilon^2(C)\right) \\
&\overset{(57)}{\leq} \overline{\phi} - \underline{\phi} + \frac{16\Upsilon^2(C)}{c_0} + \left[\frac{4\zeta_1}{\alpha c_0} + \frac{1}{c_0} + \frac{4\zeta_2}{\alpha}\right]\left(2\Upsilon^2(C)\right) \\
&= \overline{\phi} - \underline{\phi} + \frac{8\zeta_2\Upsilon^2(C)}{\alpha} + \frac{2\Upsilon^2(C)}{c_0}\left(\frac{4\zeta_1}{\alpha} + 9\right) = \bar\Gamma(x_0; C, \alpha),
\end{aligned}$$

where the last identity follows from $c_0 = 1/(1 + \|Ax^0 - b\|)$ and the definition of $\bar\Gamma(x_0; C, \alpha)$ in (55).

(c) Assume that $c_\ell \geq 4\Upsilon(C)/\eta$. This assumption, the first inequality in (57), and the fact that $c_\ell = 2c_{\ell-1}$, immediately imply that $c_{\ell-1} \geq 2\max\{\|p^{\ell-1}\|, \Upsilon(C)\}/\eta$. The statement now follows from the previous observation, line 3 of VP-ADMM, and Theorem 3.2(c) with $(y^0, q^0, c) = (x^{\ell-1}, p^{\ell-1}, c_{\ell-1})$. ∎

The next result describes the iteration-complexity of VP-ADMM in terms of total ADMM iterations (and hence BIPP calls within FP-ADMM).

Theorem 5.2 (VP-ADMM Complexity) *The following statements about* VP-ADMM *hold:*

(a) *it obtains a (ρ, η)-stationary solution of* (1)-(2) *in no more than $\log_2[1 + 4\Upsilon(C)/(c_0\eta)] + 1$ calls to* FP-ADMM*;*

(b) *its total number of* FP-ADMM *iterations (and hence* BIPP *calls within* FP-ADMM*) is bounded by*

$$\frac{\zeta_2\bar\Gamma(x^0; C, \alpha)}{\rho^2}\left[\frac{8\Upsilon(C)}{\eta} + c_0\right] + \left[1 + \frac{\zeta_1\bar\Gamma(x^0; C, \alpha)}{\rho^2}\right]\log_2\left(2 + \frac{8\Upsilon(C)}{c_0\eta}\right),$$

where (ζ_1, ζ_2), $\Upsilon(C)$ and $\bar\Gamma(x^0; C, \alpha)$ are as in (20), (33) *and* (55), *respectively, and c_0 is as in line 1 of* VP-ADMM.

Proof: (a) Assume for the sake of contradiction that VP-ADMM generates an iteration index $\hat\ell$ such that $\hat\ell > 1 + \log_2[1 + 4\Upsilon(C)/(c_0\eta)] > 1$, and hence

$$c_{\hat\ell-1} = c_0 2^{\hat\ell-1} > c_0\left(1 + \frac{4\Upsilon(C)}{c_0\eta}\right) > \frac{4\Upsilon(C)}{\eta}.$$

Using Lemma 5.1(c) with $\ell = \hat{\ell} - 1 \geq 1$, we conclude that the quadruple $(x^{\hat{\ell}-1}, p^{\hat{\ell}-1}, u^{\hat{\ell}-1}, \varepsilon_{\hat{\ell}-1})$ is a (ρ, η) stationary solution of problem (1)-(2), and hence satisfies $\|Ax^{\hat{\ell}-1} - b\| \leq \eta$. In view of line 5 of VP-ADMM, this implies that VP-ADMM stops at the $(\hat{\ell}-1)$-th iteration, which hence contradicts the fact that $\hat{\ell}$ is an iteration index. We have thus proved that (a) holds.

(b) Let $\tilde{\ell}$ denote the total number of FP-ADMM calls and observe that $\tilde{\ell} \leq 1 + \log_2[1 + 4\Upsilon(C)/(c_0\eta)]$ due to (a). Now, using Lemma 5.1(b) and the previous observation, we have that the overall number of iterations performed by FP-ADMM is bounded by

$$\begin{aligned}
\sum_{\ell=1}^{\tilde{\ell}} \left[\left(\frac{\zeta_1 + c_{\ell-1}\zeta_2}{\rho^2}\right)\bar{\Gamma}(x^0; C, \alpha) + 1\right] &= \left[1 + \frac{\zeta_1 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\right]\tilde{\ell} + \frac{\zeta_2 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\sum_{\ell=1}^{\tilde{\ell}} c_{\ell-1} \\
&\leq \left[1 + \frac{\zeta_1 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\right]\tilde{\ell} + \frac{c_0\zeta_2 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\left(2^{\tilde{\ell}} - 1\right) \\
&\leq \left[1 + \frac{\zeta_1 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\right]\tilde{\ell} + \frac{c_0\zeta_2 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\left(1 + \frac{8\Upsilon(C)}{c_0\eta}\right).
\end{aligned}$$

The result now follows by using that $\tilde{\ell} \leq 1 + \log_2[1 + 4\Upsilon(C)/(c_0\eta)]$. ∎

We now make some comments about Theorem 5.2. First, it follows from Theorem 5.2(a) that the final penalty parameter generated by VP-ADMM is $\mathcal{O}(\eta^{-1})$. Second, it follows from Theorem 5.2(a) that VP-ADMM ends with a (ρ, η)-stationary solution of (1)-(2) by calling FP-ADMM (and hence doubling the penalty parameter) no more than $\mathcal{O}(\log_2(\eta^{-1}))$ times. Third, under the mild assumptions that $\|Ax^0 - b\| = \mathcal{O}(1)$, $\alpha = \Omega(1)$, and $C = \mathcal{O}(1)$, Theorem 5.2(b) and the fact that $\zeta_2 = 0$ when $B = 1$ (see (20)), imply that the complexity of VP-ADMM, in terms of the tolerances only, is:

- $\tilde{\mathcal{O}}(\rho^{-2}\eta^{-1})$ if $B > 1$, and thus $\tilde{\mathcal{O}}(\epsilon^{-3})$;
- $\tilde{\mathcal{O}}(\rho^{-2})$ if $B = 1$, and thus $\tilde{\mathcal{O}}(\epsilon^{-2})$,

where $\epsilon := \min\{\rho, \eta\}$. On the other hand, FP-ADMM only achieves the above complexities with (a generally non-computable) $c = \Theta(\eta^{-1})$ and with the condition that $c\|Ax^0 - b\|^2 = \mathcal{O}(1)$ (see the first paragraph following Theorem 3.2), or equivalently, $\|Ax^0 - b\| = \mathcal{O}(\eta^{1/2})$, and hence the initial point x^0 being nearly feasible. Finally, the above complexity for $B = 1$ is similar to those derived for some AL methods in terms of tolerance dependencies (e.g., see [35, Theorem 2.3(b)], [58, Proposition 3.7(a)], and [64, 65]).

6 An adaptive prox stepsize VP-ADMM

This section describes an adaptive prox stepsize version of VP-ADMM, referred to as Adapt-ADMM, which requires no knowledge of the weak convexity parameters m_t's. It contains four subsections. Subsection 6.1 describes an adaptive prox stepsize version of BIPP, referred to as A-BIPP. Subsection 6.2 presents Adapt-ADMM, whose description invokes A-BIPP instead of BIPP, and shows that its iteration-complexity is similar to its constant stepsize VP-ADMM analog. Subsection 6.3 provides further details about the implementation of A-BIPP. Finally, Subsection 6.4 proves two main technical results stated in Subsection 6.1.

6.1 A-BIPP: A variable prox stepsize version of BIPP

This subsection describes A-BIPP and states two main results. The first one shows that BIPP is a special case of A-BIPP when the prox stepsize input to the former is sufficiently small. The second one, which is a generalization of Proposition 3.1 to the A-BIPP context, states the main properties of A-BIPP. The subsection ends with a proof of Proposition 3.1, which follows as a consequence of these two results.

Subroutine A-BIPP

Input: $(z, p, \lambda, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}^B_{++} \times \mathbb{R}_{++}$

Output: $(z^+, v^+, \delta_+, \lambda^+) \in \mathcal{H} \times \mathbb{R}^l \times \mathbb{R}_{++} \times \mathbb{R}^B_{++}$

1: **for** $t = 1, \ldots, B$ **do**

2: $\quad \lambda_t^+ \leftarrow \lambda_t$

3: $\quad$ compute a $(1/8; z_t)$-relative stationary solution $(z_t^+, r_t^+, \varepsilon_t^+)$ of

$$\min_{u \in \mathbb{R}^{n_t}} \left\{ \lambda_t^+ \hat{\mathcal{L}}_c(z_{<t}^+, u, z_{>t}; p) + \frac{1}{2}\|u - z_t\|^2 + \lambda_t^+ h_t(u) \right\} \tag{60}$$

$\quad$ with composite term $\lambda_t^+ h_t(\cdot)$ (see Definition 2.1)

4: $\quad$ **if** z_t^+ does **NOT** satisfy

$$\mathcal{L}_c(z_{<t}^+, z_t, z_{>t}; p) - \mathcal{L}_c(z_{<t}^+, z_t^+, z_{>t}; p) \geq \frac{1}{8\lambda_t^+}\|z_t^+ - z_t\|^2 + \frac{c}{4}\|A_t(z_t^+ - z_t)\|^2 \tag{61}$$

$\quad$ **then**

5: $\quad\quad \lambda_t^+ \leftarrow \lambda_t^+/2$ and go to line 3

6: $z^+ \leftarrow (z_1^+, \ldots, z_B^+)$ and $\lambda^+ \leftarrow (\lambda_1^+, \ldots, \lambda_B^+)$

7: **for** $t = 1, \ldots, B$ **do**

8: $\quad v_t^+ \leftarrow \nabla_t f(z_{<t}^+, z_t^+, z_{>t}^+) - \nabla_t f(z_{<t}^+, z_t^+, z_{>t}) + \dfrac{r_t^+}{\lambda_t^+} + cA_t^* \left[\sum_{s=t+1}^B A_s(z_s^+ - z_s)\right] - \dfrac{1}{\lambda_t^+}(z_t^+ - z_t)$

9: $v^+ \leftarrow (v_1^+, \ldots, v_B^+)$ and $\delta_+ \leftarrow (\varepsilon_1^+/\lambda_1^+) + \ldots + (\varepsilon_B^+/\lambda_B^+)$

10: **return** $(z^+, v^+, \delta_+, \lambda^+)$

We now make two remarks about the way A-BIPP computes the prox stepsize λ_t^+ for every $t \in \{1, \ldots, B\}$. First, in contrast to BIPP which keeps λ_t constant, A-BIPP adaptively searches for a suitable λ_t^+ in the loop consisting of lines 2 to 5, referred to as the t-th A-BIPP loop in our discussion below.

Each iteration of the t-th A-BIPP loop halves λ_t^+ and the loop terminates when a prox stepsize λ_t^+ satisfying (61) is generated. Second, if $(z^+, v^+, \delta_+, \lambda^+) = \text{A-BIPP}(z, p, \lambda, c)$ and $\lambda^+ = \lambda$, then $(z^+, v^+, \delta_+) = \text{BIPP}(z, p, \lambda, c)$; hence, if the initial and final prox stepsizes of A-BIPP are the same, then both BIPP and A-BIPP are equivalent.

The following result shows that BIPP is a special case of A-BIPP when the prox stepsize input to the former is sufficiently small.

Proposition 6.1 *Assume that $1/\lambda_t \in [2m_t, \infty)$ for every $t = 1, \ldots, B$ and $(z^+, v^+, \delta_+) = \text{BIPP}(z, p, \lambda, c)$ for some $(z, p, \lambda, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}^B_{++} \times \mathbb{R}_{++}$. Then, the following statements hold:*

(a) for each $t \in \{1, \ldots, B\}$, the smooth part of the objective function of the t-th block subproblem (60) is $(1/2)$-strongly convex;

(b) $(z^+, v^+, \delta_+, \lambda) = \text{A-BIPP}(z, p, \lambda, c)$.

Proposition 6.1(b) shows that the reverse of the implication in the second remark preceding Proposition 6.1 holds if λ input to BIPP is sufficiently small.

The following result, which is a more general version of Proposition 3.1, describes the main properties of the quadruple $(z^+, v^+, \delta_+, \lambda^+)$ output by A-BIPP.

Proposition 6.2 *For a given $(z, p, \lambda, c) \in \mathcal{H} \times A(\mathbb{R}^n) \times \mathbb{R}^B_{++} \times \mathbb{R}_{++}$, the following statements about* A-BIPP *with input (z, p, λ, c) hold:*

(a) for every $t \in \{1, \ldots, B\}$, the t-th A-BIPP *loop stops in at most $1 + \lceil \log_2(1 + 2m_t\lambda_t) \rceil$ iterations, and as a consequence* A-BIPP *terminates; moreover, the prox stepsize λ^+ output by* A-BIPP *satisfies*

$$\lambda_t^+ \leq \lambda_t, \quad \max\left\{\frac{1}{\lambda_t}, 4m_t\right\} = \max\left\{\frac{1}{\lambda_t^+}, 4m_t\right\} \quad \forall t = 1, \ldots, B; \tag{62}$$

(b) if $(z^+, v^+, \delta_+, \lambda^+) = \text{A-BIPP}(z, p, \lambda, c)$, then inclusion (23) holds and

$$\|v^+\|^2 + \delta_+ \leq \left[1 + \frac{50}{\min(\lambda^+)} + 48L^2 \max(\lambda^+) + c\zeta_2\right] \left[\mathcal{L}_c(z; p) - \mathcal{L}_c(z^+; p)\right], \tag{63}$$

where ζ_2 is as in (20) and L is as in (21).

We now use Propositions 6.1 and 6.2 to prove Proposition 3.1.

Proof of Proposition 3.1: Assume that $(z^+, v^+, \delta_+) = \text{BIPP}(z, p, \lambda, c)$ and λ is chosen as in (22), and hence that $1/\lambda_t \in [2m_t, \infty)$ for $t = 1, \ldots, B$. It then follows from Proposition 6.1(a) that Proposition 3.1(a) holds. Moreover, Proposition 6.1(b) implies that $(z^+, v^+, \delta_+, \lambda) = \text{A-BIPP}(z, p, \lambda, c)$, and hence that the assumption of Proposition 6.2 holds with $\lambda^+ = \lambda$. The conclusion of Proposition 6.2(b) then implies that inclusion (23), and inequality (63) with λ^+ replaced by λ hold, i.e.,

$$\|v^+\|^2 + \delta_+ \le \left[1 + \frac{50}{\min(\lambda)} + 48L^2 \max(\lambda) + c\zeta_2\right] \left[\mathcal{L}_c(z;p) - \mathcal{L}_c(z^+;p)\right].$$

Using the above inequality, the definition of ζ_1, and the fact that (22) implies that

$$\frac{1}{\min(\lambda)} = \max_{1\le t\le B} \frac{1}{\lambda_t} \overset{(22)}{=} 2\max\left\{1, \max_{1\le t\le B} m_t\right\}, \qquad \max(\lambda) = \max_{1\le t\le B} \lambda_t \overset{(22)}{\le} \frac{1}{2},$$

we then conclude that (24) also holds. We have thus proved that Proposition 3.1(b) holds. ∎

6.2 An adaptive prox stepsize VP-ADMM

This subsection presents Adapt-ADMM, an adaptive prox stepsize analog of VP-ADMM, and argues that the complexity results for VP-ADMM derived in Section 5 can be naturally extended to Adapt-ADMM, using the observations made in the paragraph preceding Lemma 4.2.

We start by stating Adapt-ADMM, the adaptive prox stepsize analog of VP-ADMM. In contrast to VP-ADMM, which uses FP-ADMM as a subroutine, the extended description below incorporates all the details of the adaptive FP-ADMM analog into a single loop, i.e., the second one.

Algorithm 3 Adapt-ADMM ($m = (m_1, \ldots, m_B)$ is **NOT** required)

Universal Input: $(\rho, \eta) \in \mathbb{R}^2_{++}$, $\alpha > 0$, $C \ge \rho$
Input: $x^0 \in \mathcal{H}$ and $\gamma^0 \in \mathbb{R}^B_{++}$
Output: $(\hat{x}, \hat{p}, \hat{u}, \hat{\varepsilon}, \hat{c})$ that satisfies the conclusions of Theorem 6.3.

1: $p^0 = (p_1^0, \ldots, p_B^0) \leftarrow (0, \ldots, 0)$, $c_0 \leftarrow 1/[1 + \|Ay^0 - b\|]$

2: **for** $\ell \leftarrow 1, 2, \ldots$ **do**
3: $\quad T_0 \leftarrow 0$, $k \leftarrow 0$, $c \leftarrow c_{\ell-1}$, $(y^0, q^0, \lambda^0, c) \leftarrow (x^{\ell-1}, p^{\ell-1}, \gamma^{\ell-1}, c_{\ell-1})$
4: $\quad$ **for** $i \leftarrow 1, 2, \ldots$ **do**
5: $\quad\quad (y^i, v^i, \delta_i, \lambda^i) = \text{A-BIPP}(y^{i-1}, q^{i-1}, \lambda^{i-1}, c)$
6: $\quad\quad$ **if** $\|v^i\|^2 + \delta_i \le \rho^2$ **then**
7: $\quad\quad\quad k \leftarrow k + 1,\ i_k \leftarrow i$
8: $\quad\quad\quad q^i \leftarrow q^{i-1} + c(Ay^i - b)$
9: $\quad\quad\quad (x^\ell, p^\ell, u^\ell, \varepsilon_\ell, \gamma^\ell) = (y^i, q^i, v^i, \delta_i, \lambda^i)$ and **go to line 16**

10: $\quad\quad T_i = \mathcal{L}_c(y^{i-1}; q^{i-1}) - \mathcal{L}_c(y^i; q^{i-1}) + T_{i-1}$
11: $\quad\quad$ **if** $\|v^i\|^2 + \delta_i \le C^2$ and $\dfrac{\rho^2}{\alpha(k+1)} \ge \dfrac{T_i}{i}$ **then**
12: $\quad\quad\quad k \leftarrow k + 1,\ i_k \leftarrow i$
13: $\quad\quad\quad q^i \leftarrow q^{i-1} + c(Ay^i - b)$
14: $\quad\quad$ **else**
15: $\quad\quad\quad q^i \leftarrow q^{i-1}$

16: $\quad c_\ell \leftarrow 2c_{\ell-1}$
17: $\quad$ **if** $\|Ax^\ell - b\| \le \eta$ **then**
18: $\quad\quad$ **return** $(\hat{x}, \hat{p}, \hat{u}, \hat{\varepsilon}, \hat{c}) = (x^\ell, p^\ell, u^\ell, \varepsilon_\ell, c_\ell)$

We now make some comments about Adapt-ADMM. First, Adapt-ADMM consists of two main loops. The outer one, indexed by ℓ, starts in line 2 and ends in line 18. The inner one, indexed by i, starts in line 4 and ends in line 15. The iterations of an inner loop, referred simply to as an *inner iteration*, execute an adaptive variant of FP-ADMM, with BIPP replaced by its adaptive counterpart A-BIPP (see line 5 of Adapt-ADMM). Second, an iteration of the outer loop, referred to as an *outer iteration*, can be viewed as an adaptive version of a VP-ADMM iteration. Third, in contrast to VP-ADMM, which uses the constant prox stepsize (22), Adapt-ADMM chooses the sequence of inner and outer prox stepsizes $\{\lambda^i\}$ and $\{\gamma^\ell\}$ adaptively due to its call to A-BIPP in line 5. Thus, Adapt-ADMM is an ADMM variant which is fully adaptive, i.e., adaptive relative to all instance parameters associated with problem (1) and hence fulfills the list of attributes listed in the "Contributions" part of the Introduction.

We next discuss how the iteration-complexity of Adapt-ADMM can be established by following an argument close to, but slightly different than, the one used for FP-ADMM/VP-ADMM. We start with some remarks about the sequence of inner prox stepsizes $\{\lambda^i = (\lambda_1^i, \ldots, \lambda_B^i)\}$ generated within an arbitrary inner loop of Adapt-ADMM. First, it is straightforward to observe that line 5 of Adapt-ADMM and Proposition 6.2 with $(z, p, \lambda, c) = (y^{i-1}, q^{i-1}, \lambda^{i-1}, c)$, imply that, for every inner iteration i of any inner loop, we have

$$\lambda_t^i \le \lambda_t^{i-1}, \quad \max\left\{\frac{1}{\lambda_t^i}, 4m_t\right\} = \max\left\{\frac{1}{\lambda_t^{i-1}}, 4m_t\right\} \quad \forall t \in \{1, \ldots, B\}, \tag{64}$$

$$v^i \in \nabla f(y^i) + \partial_{\delta_i} h(y^i) + A^*[q^{i-1} + c(Ay^i - b)], \tag{65}$$

$$\|v^i\|^2 + \delta_i \le \left[1 + \frac{50}{\min(\lambda^i)} + 48L^2 \max(\lambda^i) + c\zeta_2\right]\left[\mathcal{L}_c(y^{i-1}; q^{i-1}) - \mathcal{L}_c(y^i; q^{i-1})\right], \tag{66}$$

where ζ_2 is as in (20). Relation (64) immediately implies that

$$\lambda_t^i \le \gamma_t^0, \quad \max\left\{\frac{1}{\lambda_t^i}, 4m_t\right\} = \max\left\{\frac{1}{\gamma_t^0}, 4m_t\right\} \quad \forall t \in \{1, \ldots, B\}. \tag{67}$$

Now, (66) and (67) imply that every inner iteration i of any inner loop satisfies

$$\|v^i\|^2 + \delta_i \le [\zeta_1 + c\zeta_2]\left[\mathcal{L}_c(y^{i-1}; q^{i-1}) - \mathcal{L}_c(y^i; q^{i-1})\right], \tag{68}$$

where the pair (ζ_1, ζ_2) in this subsection is defined as

$$\zeta_1 = 1 + 50\bar{\chi} + 48L^2 \max(\gamma^0), \quad \zeta_2 = 24(B-1)\|A\|_\dagger^2, \tag{69}$$

and

$$\bar{\chi} := \max_{1 \le t \le B}\left(\max\left\{\frac{1}{\gamma_t^0}, 4m_t\right\}\right).$$

Relations (65) and (68) then imply that condition **C1** in Section 4 holds with (ζ_1, ζ_2) as in (69). Thus, as observed in the paragraph containing condition **C1**, all the results derived in Sections 4 and 5, except for Lemma 4.1, hold. In particular, Theorem 5.2 translated to the context of Adapt-ADMM becomes as follows:

Theorem 6.3 (Adapt-ADMM Complexity) *The following statements about* Adapt-ADMM *hold:*

(a) it obtains a (ρ, η)-stationary solution of (1)-(2) in no more than $\log_2[1 + 4\Upsilon(C)/(c_0\eta)] + 1$ outer iterations;

(b) the total number of inner iterations performed by it is bounded by

$$\frac{\zeta_2 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\left[\frac{8\Upsilon(C)}{\eta} + c_0\right] + \left[1 + \frac{\zeta_1 \bar{\Gamma}(x^0; C, \alpha)}{\rho^2}\right]\log_2\left(2 + \frac{8\Upsilon(C)}{c_0\eta}\right),$$

where (ζ_1, ζ_2), $\Upsilon(C)$ and $\bar{\Gamma}(x^0; C, \alpha)$ are as in (69), (32) and (55), respectively, and c_0 is as in line 1 of Adapt-ADMM.

Under the mild assumptions that $\|Ax^0 - b\| = \mathcal{O}(1)$, $\alpha = \Omega(1)$, and $C = \mathcal{O}(1)$, Theorem 6.3(b) implies that the complexity of Adapt-ADMM, in terms of the tolerances only, is $\tilde{\mathcal{O}}(\epsilon^{-3})$ when $B > 1$, and $\tilde{\mathcal{O}}(\epsilon^{-2})$ when $B = 1$, where $\epsilon := \min\{\rho, \eta\}$.

6.3 Further implementation issues of an A-BIPP loop

This subsection addresses some aspects related to the computation of a $(1/8, z_t)$-relative stationary solution of the t-th block subproblem (60) in line 3 of A-BIPP.

We first recall that throughout our presentation in Section 6, we have assumed that a $(1/8, z_t)$-relative stationary solution of (60) can be obtained every time line 3 of A-BIPP is executed. Such an assumption is reasonable if an exact solution z_t^+ of (60) can be computed in closed form since then $(z_t^+, v_t^+, \varepsilon_t^+) = (z_t^+, 0, 0)$ is a $(1/8; z_t)$-relative stationary solution of (60).

We now discuss the issues of finding a $(1/8, z_t)$-relative stationary solution of (60) using the ADAP-FISTA described in Appendix B with input $(\mu_0, M_0) = (1/2, \lambda_t c\|A_t\|^2)$, and hence with the same input as in the discussion on the second last paragraph of Subsection 3.1. Recall that in that paragraph, as well as in here, we assume that $\nabla_t f(x_1, \ldots, x_B)$ is $\tilde{L}_t$-Lipschitz continuous with respect to x_t. If $\lambda_t^+ > 1/(2m_t)$ in line 3, then ADAP-FISTA may not be able to find the required relative stationary solution. This is because the smooth part of the objective function of (60) with the above input is not necessarily $(1/2)$-strongly convex (see statement (a) of Lemma 6.4 below), which can cause failure of ADAP-FISTA (see Proposition B.1(c) with $\mu_0 = 1/2$). Nevertheless, regardless of whether ADAP-FISTA succeeds or fails, a similar reasoning as in the second last paragraph of Subsection 3.1 shows that it terminates in $\mathcal{O}([\lambda_t^+(\tilde{L}_t + c\|A_t\|^2)]^{1/2})$ iterations. Moreover, failure of ADAP-FISTA signals that the current prox stepsize λ_t^+ is too large. In this case, λ_t^+ should be halved regardless of whether (61) is satisfied or not. An argument close to the one used in the proof of Proposition 6.2(a), which not only uses Lemma 6.4(b) but also Lemma 6.4(a), shows that this slightly modified version of A-BIPP terminates in at most $1 + \lceil \log_2(1 + 4m_t\lambda_t)\rceil$ loop iterations.

6.4 Proofs of Propositions 6.1 and 6.2

This subsection contains the proof of Propositions 6.1 and 6.2. First, we present a technical result, which states that if the t-th prox stepsize input for A-BIPP is sufficiently small, then its t-th loop terminates.

The next result uses the fact that line 3 of A-BIPP, together with Definition 2.1, generates a triple $(z_t^+, r_t^+, \varepsilon_t^+) \in \mathcal{H} \times \mathbb{R}^l \times \mathbb{R}_+$ such that

$$\begin{aligned} r_t^+ &\in \nabla\left[\lambda_t^+ \hat{\mathcal{L}}_c(z_{<t}^+, \cdot, z_{>t}; p) + \frac{1}{2}\|\cdot - z_t\|^2\right](z_t^+) + \partial_{\varepsilon_t^+}(\lambda_t^+ h_t)(z_t^+), \\ &\|r_t^+\|^2 + 2\varepsilon_t^+ \le \frac{1}{8}\|z_t^+ - z_t\|^2, \end{aligned} \tag{70}$$

where $\hat{\mathcal{L}}_c(\cdot; p)$ is as in (17).

Lemma 6.4 *Let $t \in \{1, \ldots, B\}$ be given and assume that the prox stepsize λ_t^+ at a certain iteration of the t-th* A-BIPP *loop satisfies $1/\lambda_t^+ \in [2m_t, \infty)$. Then, the following statements hold for this iteration of the t-th loop:*

(a) the smooth part of the objective function of the t-th block subproblem (60) *is $(1/2)$-strongly convex;*

(b) the t-th loop ends at this iteration.

Proof: (a) The assumption that $1/\lambda_t^+ \in [2m_t, \infty)$ implies that the matrix $B_t := (1 - \lambda_t^+ m_t)I + \lambda_t^+ c A_t^* A_t$ is clearly positive definite, and hence defines the norm $\|\cdot\|_{B_t}$ whose square is

$$\|\cdot\|_{B_t}^2 := \langle\, \cdot\,,\, B_t(\cdot)\,\rangle \ge \lambda_t^+ c\|A_t(\cdot)\|^2 + \frac{1}{2}\|\cdot\|^2. \tag{71}$$

Now, let $\Psi_t(\cdot)$ denote the smooth part of the objective function of the t-th block subproblem (60), i.e.,

$$\Psi_t(\cdot) := \lambda_t^+ \hat{\mathcal{L}}_c(z_{<t}^+, \cdot, z_{>t}; p) + \frac{1}{2}\|\cdot - z_t\|^2 \tag{72}$$

where $\hat{\mathcal{L}}_c(\cdot\,; p)$ is as in (17). Moreover, using assumption (A2), the above definitions of B_t and the norm $\|\cdot\|_{B_t}$, the definitions of $\hat{\mathcal{L}}_c(\cdot\,;\cdot)$ and $\Psi_t(\cdot)$ in (17) and (72), respectively, we easily see that the function $\Psi_t(\cdot) - \frac{1}{2}\|\cdot\|_{B_t}^2$ is convex, and hence $\Psi_t(\cdot)$ is $(1/2)$-strongly convex due to the inequality in (71).

(b) It suffices to show that (61) holds at this loop iteration. Due to (72) and the inclusion in (70), the quadruple $(z_t^+, r_t^+, \varepsilon_t^+, \lambda_t^+)$ satisfies

$$r_t^+ \overset{(70)}{\in} \nabla \Psi_t(z_t^+) + \partial_{\varepsilon_t^+}(\lambda_t^+ h_t)(z_t^+) \subset \partial_{\varepsilon_t^+}\left[\Psi_t + \lambda_t^+ h_t\right](z_t^+)$$

where the set inclusion is due to [23, Thm. 3.1.1 of Ch. XI], and the fact that $\Psi_t(\cdot)$ and $\lambda_t^+ h_t(\cdot)$ are convex functions. Since $\Psi_t(\cdot) - \frac{1}{2}\|\cdot\|_{B_t}^2$ is convex, it follows from Lemma A.4 with $\psi = \Psi_t + \lambda_t^+ h_t$, $(\xi, \tau, Q) = (1, 1, B_t)$, $(u, y, v) = (z_t, z_t^+, r_t^+)$, and $\eta = \varepsilon_t^+$, that

$$\begin{aligned}
&\lambda_t^+ \mathcal{L}_c(z_{<t}^+, z_t, z_{>t}; p) - \left[\lambda_t^+ \mathcal{L}_c(z_{<t}^+, z_t^+, z_{>t}; p) + \frac{1}{2}\|z_t^+ - z_t\|^2\right] \\
&= \left[\Psi_t(z_{<t}^+, z_t, z_{>t}; p) + \lambda_t^+ h_t(z_t)\right] - \left[\Psi_t(z_{<t}^+, z_t^+, z_{>t}; p) + \lambda_t^+ h_t(z_t^+)\right] \\
&\geq \frac{1}{4}\|z_t^+ - z_t\|_{B_t}^2 - 2\varepsilon_t^+ + \langle r_t^+, z_t - z_t^+\rangle \overset{(71)}{\geq} \frac{\lambda_t^+ c}{4}\|A_t(z_t^+ - z_t)\|^2 - 2\varepsilon_t^+ + \langle r_t^+, z_t - z_t^+\rangle,
\end{aligned}$$

where the first equality follows from (4), (17), and (72), the first inequality is due to Lemma A.4, and the last inequality is due to (71). Using the previous inequality, the inequality $ab \leq (a^2 + b^2)/2$ with $(a, b) = (\sqrt{2}\|r_t^+\|, (1/\sqrt{2})\|z_t^+ - z_t\|)$, and the inequality in (70), we conclude that

$$\begin{aligned}
&\mathcal{L}_c(z_{<t}^+, z_t, z_{>t}; p) - \mathcal{L}_c(z_{<t}^+, z_t^+, , z_{>t}; p) \\
&\qquad \geq \frac{1}{2\lambda_t^+}\|z_t^+ - z_t\|^2 + \frac{c}{4}\|A_t(z_t^+ - z_t)\|^2 - \frac{1}{\lambda_t^+}\left(\|\sqrt{2}r_t^+\| \left\|\frac{1}{\sqrt{2}}(z_t^+ - z_t)\right\| + 2\varepsilon_t^+\right) \\
&\qquad \geq \frac{1}{2\lambda_t^+}\|z_t^+ - z_t\|^2 + \frac{c}{4}\|A_t(z_t^+ - z_t)\|^2 - \frac{1}{\lambda_t^+}\left(\|r_t^+\|^2 + \frac{1}{4}\|z_t^+ - z_t\|^2 + 2\varepsilon_t^+\right) \\
&\qquad \overset{(70)}{\geq} \frac{1}{2\lambda_t^+}\|z_t^+ - z_t\|^2 + \frac{c}{4}\|A_t(z_t^+ - z_t)\|^2 - \frac{1}{\lambda_t^+}\left(\frac{1}{4} + \frac{1}{8}\right)\|z_t^+ - z_t\|^2 \\
&\qquad = \frac{1}{8\lambda_t^+}\|z_t^+ - z_t\|^2 + \frac{c}{4}\|A_t(z_t^+ - z_t)\|^2,
\end{aligned}$$

and hence that (61) holds. By the logic of A-BIPP, it then follows that the t-th A-BIPP loop terminates at the current loop iteration with λ_t^+ being the final t-th stepsize output by A-BIPP. We have thus proved that (b) holds. ∎

It follows from Lemma 6.4(b) that, if function f restricted to its t-th block variable is convex, i.e., $m_t = 0$, then the t-th A-BIPP loop terminates in one iteration with $\lambda_t^+ = \lambda_t$. Hence, A-BIPP does not update λ_t when $m_t = 0$.

Proposition 6.1 now follows immediately from Lemma 6.4. Indeed, both of its statements follow from its assumptions, line 2 of A-BIPP, and Lemma 6.4 with $\lambda_t^+ = \lambda_t$ for every $t \in \{1, \ldots, B\}$.

We are now ready to prove Proposition 6.2

Proof of Proposition 6.2: (a) Let $t \in \{1, \ldots, B\}$ be given. Recall from Lemma 6.4(b) that if $1/\lambda_t^+ \in [2m_t, \infty)$ at some iteration of the t-th A-BIPP loop, then the loop must terminate at this iteration. Using this observation and the fact that $1/\lambda_t^+$ is doubled every time line 5 of A-BIPP is executed, we easily see that the t-th A-BIPP loop stops in at most $1 + \lceil \log_2(1 + 2m_t\lambda_t) \rceil$ iterations, the inequality in (62) holds, and

$$\frac{1}{\lambda_t^+} \leq \max\left\{\frac{1}{\lambda_t}, 4m_t\right\}.$$

The identity in (62) now follows immediately from the inequality $1/\lambda_t \leq 1/\lambda_t^+$ and the one above.

(b) We first prove the inclusion in (23). Using the inclusion in (70), the definition of $\hat{\mathcal{L}}_c(\cdot\,; p)$ in (17), and relation (11) with $(\varepsilon, \beta) = (\varepsilon_t^+, \lambda_t^+)$, we easily see that

$$\begin{aligned}
\frac{r_t^+}{\lambda_t^+} &\overset{(70)}{\in} \nabla_t f(z_{<t}^+, z_t^+, z_{>t}) + A_t^*\left[p + c[A(z_{<t}^+, z_t^+, z_{>t}) - b]\right] + \frac{1}{\lambda_t^+}(z_t^+ - z_t) + \partial_{(\varepsilon_t^+/\lambda_t^+)} h_t(z_t^+) \\
&= \nabla_t f(z_{<t}^+, z_t^+, z_{>t}) + A_t^*\left(p + c(Az^+ - b) - c\sum_{s=t+1}^{B} A_s(z_s^+ - z_s)\right) + \frac{1}{\lambda_t^+}(z_t^+ - z_t) + \partial_{(\varepsilon_t^+/\lambda_t^+)} h_t(z_t^+),
\end{aligned}$$

for every $t \in \{1, \ldots, B\}$. Rearranging the above inclusion and using the definition of v_t^+ in line 8 of A-BIPP, we have

$$v_t^+ \in \nabla_t f(z^+) + \partial_{(\varepsilon_t^+/\lambda_t^+)} h_t(z_t^+) + A_t^* \left[p + c(Az^+ - b)\right] \quad \forall t \in \{1, \ldots, B\}.$$

Moreover, we have that $\partial_{(\varepsilon_1^+/\lambda_1^+)} h_1(z_1^+) \times \ldots \times \partial_{(\varepsilon_B^+/\lambda_B^+)} h_B(z_B^+) \subseteq \partial_{\delta_+} h(z^+)$, due to (12) and the definition of δ_+ in line 9 of A-BIPP. The inclusion in (23) now follows from the last two conclusions and the definition of v^+ in line 9 of A-BIPP.

We now prove the inequality in (63). Lemma 6.4(b) and the logic of A-BIPP imply that the t-th A-BIPP loop terminates with a quadruple $(z_t^+, r_t^+, \varepsilon_t^+, \lambda_t^+)$ satisfying (61). Adding (61) from $t = 1$ to $t = B$, we have

$$\Delta \mathcal{L}_c := \mathcal{L}_c(z; p) - \mathcal{L}_c(z^+; p) \geq \frac{1}{8} \sum_{t=1}^{B} \frac{\|z_t^+ - z_t\|^2}{\lambda_t^+} + \frac{c}{4} \sum_{t=1}^{B} \|A_t(z_t^+ - z_t)\|^2. \tag{73}$$

Now, using (70), (73), and that $1/\lambda_t^+ \leq 1/\min(\lambda^+)$ (see (9)), we have

$$\begin{aligned}
\sum_{t=1}^{B} \left(2\frac{\|r_t^+\|^2}{(\lambda_t^+)^2} + \frac{\varepsilon_t^+}{\lambda_t^+}\right) &\leq \left[\frac{2}{\min(\lambda^+)} + 1\right] \sum_{t=1}^{B} \left(\frac{\|r_t^+\|^2 + \varepsilon_t^+}{\lambda_t^+}\right) \\
&\overset{(70)}{\leq} \left[\frac{2}{\min(\lambda^+)} + 1\right] \sum_{t=1}^{B} \left(\frac{\|z_t^+ - z_t\|^2}{8\lambda_t^+}\right) \overset{(73)}{\leq} \left[\frac{2}{\min(\lambda^+)} + 1\right] \Delta \mathcal{L}_c.
\end{aligned}$$

Defining $D_t := \|v_t^+ - r_t^+/\lambda_t^+\|^2$, using the previous inequality, the definition of δ_+ (see line 9 of A-BIPP), and that $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, for any $a, b \in \mathbb{R}^n$, we have

$$\|v^+\|^2 + \delta_+ = \sum_{t=1}^{B} \left(\|v_t^+\|^2 + \frac{\varepsilon_t^+}{\lambda_t^+}\right) \leq \sum_{t=1}^{B} \left(2D_t + 2\frac{\|r_t^+\|^2}{(\lambda_t^+)^2} + \frac{\varepsilon_t^+}{\lambda_t^+}\right) \leq 2\sum_{t=1}^{B} D_t + \left[\frac{2}{\min(\lambda^+)} + 1\right] \Delta \mathcal{L}_c. \tag{74}$$

We will now bound $\sum_{t=1}^{B} D_t$. Using (73) and the inequality $\lambda_s^+ \leq \max(\lambda^+)$, we have

$$\|z_{>t}^+ - z_{>t}\|^2 = \sum_{s=t+1}^{B} \|z_s^+ - z_s\|^2 \leq \left(\max(\lambda^+) \sum_{s=t+1}^{B} \frac{\|z_s^+ - z_s\|^2}{\lambda_s^+}\right) \overset{(73)}{\leq} 8 \max(\lambda^+) \Delta \mathcal{L}_c. \tag{75}$$

Moreover, the definitions of D_t given above and v_t^+ in line 8 of A-BIPP, the Cauchy-Schwarz inequality, assumption (A3), and relations (73) and (75), imply that

$$\begin{aligned}
D_t &= \left\|v_t^+ - \frac{r_t^+}{\lambda_t^+}\right\|^2 = \left\|\nabla_t f(z_{\leq t}^+, z_{>t}^+) - \nabla_t f(z_{\leq t}^+, z_{>t}) + cA_t^* \left[\sum_{s=t+1}^{B} A_s(z_s^+ - z_s)\right] - \frac{(z_t^+ - z_t)}{\lambda_t^+}\right\|^2 \\
&\leq 3 \left\{\|\nabla_t f(z_{\leq t}^+, z_{>t}^+) - \nabla_t f(z_{\leq t}^+, z_{>t})\|^2 + \left[c\|A_t\| \sum_{s=t+1}^{B} \|A_s(z_s^+ - z_s)\|\right]^2 + \frac{\|z_t^+ - z_t\|^2}{(\lambda_t^+)^2}\right\} \\
&\overset{(13)}{\leq} 3 \left\{(L_{>t})^2 \|z_{>t}^+ - z_{>t}\|^2 + c^2\|A_t\|^2 (B-t) \sum_{s=t+1}^{B} \|A_s(z_s^+ - z_s)\|^2 + \frac{1}{\min(\lambda^+)} \frac{\|z_t^+ - z_t\|^2}{\lambda_t^+}\right\} \\
&\overset{(73),(75)}{\leq} 3 \left\{\left[8\max(\lambda^+)(L_{>t})^2 + 4c\|A_t\|^2(B-1)\right] \Delta \mathcal{L}_c + \frac{1}{\min(\lambda^+)} \frac{\|z_t^+ - z_t\|^2}{\lambda_t^+}\right\}.
\end{aligned}$$

The above inequality for $t = 1, \ldots, B$, (73), and the definitions of L and $\|A\|_\dagger$ in (21), then imply that

$$\begin{aligned}
\sum_{t=1}^{B} D_t &\overset{(21)}{\leq} \left[12c\|A\|_\dagger^2(B-1) + 24\max(\lambda^+)L^2\right] \Delta \mathcal{L}_c + \frac{3}{\min(\lambda^+)} \sum_{t=1}^{B} \frac{\|z_t^+ - z_t\|^2}{\lambda_t^+} \\
&\overset{(73)}{\leq} 12 \left[c\|A\|_\dagger^2(B-1) + 2\max(\lambda^+)L^2 + \frac{2}{\min(\lambda^+)}\right] \Delta \mathcal{L}_c.
\end{aligned}$$

This inequality, relation (74), and the definition of ζ_2 in (20), show that the inequality in (63) holds. ∎

7 Numerical Experiments

This section showcases the numerical performance of Adapt-ADMM on three linearly constrained non-convex (weakly convex) programming problems. Subsections 7.1 and 7.3 focus on a quadratic problem, while Subsection 7.2 focuses on the distributed *Cauchy loss* function [39]. Subsections 7.1 and 7.2 employ fewer blocks, each with a wide dimensional range, whereas Subsection 7.3 uses a large number of one-dimensional blocks. These three proof-of-concept experiments indicate that Adapt-ADMM may not only substantially outperform the relevant benchmarking methods in practice, but also be relatively robust to the relationship between block counts and sizes.

To provide an adequate benchmark for our methods, we compare six algorithmic variants in the tables presented in the following subsections. All variants follow the same core Adapt-ADMM framework, differing only in the use of adaptive or constant prox stepsizes and in the presence or absence of Lagrange multiplier updates:

- Adapt-ADMM / VP-ADMM: the original method with adaptive or constant prox stepsizes.
- Adapt-Penalty / Const-Penalty: penalty-only variants with no multiplier updates.
- Adapt-vADMM / Const-vADMM: *vanilla* ADMM variants with multiplier updates at each iteration.

It is worth mentioning that all nonadaptive variants use BIPP as their subroutine, while all adaptive variants use A-BIPP. The iteration counter reported in the tables corresponds to the total number of BIPP/A-BIPP calls, which corresponds to the total number of iterations mentioned in Theorem 5.2 for the nonadaptive variant (and hence counts BIPP calls) and in Theorem 6.3 for the adaptive variant (and hence counts A-BIPP calls).

For each of these variants, the total number of iterations ("Iter"), total runtime ("Time"), and the objective value at the solution ("$f + h$") are included in the tables presented in the following subsections. The tables also contain a column labeled "Mults" for the variants Adapt-ADMM and VP-ADMM, indicating the total number of Lagrange multiplier updates performed by the method. This column is omitted for Penalty and vADMM variants, since the number of multipliers they perform is naturally clear.

Notably, we attempted to implement the algorithm proposed in [31] using different parameter choices (θ, χ) that theoretically should ensure convergence; see (5) and (7). However, since none of these choices managed to find the desired point within the iteration limit, we omitted them from our benchmarks.

In all three subsections, we assume that the blocks for the generated instances of (1)-(2) have the same size $\bar{n}$, i.e.,

$$\bar{n} = n_1 = \ldots = n_t,$$

and hence that $n = \bar{n}B$. So, the sizes of instances are determined by a triple $(B, \bar{n}, l)$ where l is the number of rows of the constraint matrix A. Instances with $\bar{n} < l$ are usually harder to solve since they further "deviate" from *the last block condition* described in the Introduction (see paragraph following (7)).

The parameters $(\omega, B, \bar{n}, l)$ are specified at the beginning of each table. Each matrix $A_i \in \mathbb{R}^{l\times\bar{n}}$, corresponding to the linear constraint in each problem, is filled with i.i.d. standard-normal entries. The penalty parameter and the Lagrange multiplier are chosen in accordance with line 1 of VP-ADMM, i.e., $c_0 = 1/(1 + \|Ax^0 - b\|)$ and $p^0 = \mathbf{0}$. Moreover, we fix the input C as $C = 10^3\rho(1 + \|\nabla f(x^0)\|)$. Finally, for consistency, all of the variants with adaptive stepsizes were initialized using the same value, setting $\lambda_i^0 = 10$ for each block.

For each of the problems in Subsections 7.1 and 7.2, we solve the corresponding subproblem (19) using the ADAP-FISTA routine described in Appendix B, with $(M_0, \beta, \mu, \chi) = (1, 1.2, 0.5, 0.001)$. The routine terminates once it produces a $(1/8, z_t)$-stationary point. If this criterion is not met, the current stepsize λ_t is halved, and ADAP-FISTA is restarted with the reduced stepsize. Since the subproblems in Subsection 7.3 are all one-dimensional, they are solved exactly without invoking ADAP-FISTA.

All algorithms executed were run for a maximum of 100,000 iterations. Any algorithm reaching this limit required at least 10 milliseconds to complete. A method is considered to outperform another when it achieves both a lower iteration count and a shorter total runtime.

To ensure timely execution, each algorithm was terminated upon reaching the above iteration limit or upon finding an approximate stationary triple (x^+, p^+, v^+) satisfying the relative error criterion

$$v^+ \in \nabla f(x^+) + \partial h(x^+) + A^* p^+, \quad \frac{\|v^+\|}{1 + \|\nabla f(x^0)\|} \le \rho, \quad \frac{\|Ax^+ - b\|}{1 + \|Ax^0 - b\|} \le \eta,$$

with $\rho = \eta = 10^{-5}$.

All experiments were implemented and executed in MATLAB 2024b and run on a macOS machine with an Apple M3 Max chip (14 Cores), and 96 GB of memory. For the sake of brevity, our benchmark only considers randomly generated dense instances. Its main goal is to demonstrate that the ADMMs presented in this work are promising.

7.1 Nonconvex Distributed Quadratic Programming (DQP) Problem

This subsection studies the performance of the ADMM variants for finding stationary points of a nonconvex block distributed quadratic programming problem with B blocks (DQP).

For a given pair $(l, \bar{n}) \in \mathbb{N}^2_{++}$ with $l < \bar{n}B$, the B-block DQP is formulated as

$$\min_{x=(x_1,\ldots,x_B)\in\mathbb{R}^{\bar{n}B}} \left\{ \sum_{i=1}^{B} \left[\frac{1}{2}\langle x_i, P_i x_i\rangle + \langle x_i, r_i\rangle \right] \ : \ \|x_i\|_1 \le \omega, \, \forall i = 1, \ldots B, \text{ and } \sum_{i=1}^{B} A_i x_i = b \right\},$$

where $\omega > 0$, $b \in \mathbb{R}^l$, $P_i \in \mathbb{R}^{\bar{n}\times\bar{n}}$ is a symmetric indefinite matrix, $x_i \in \mathbb{R}^{\bar{n}}$, $r_i \in \mathbb{R}^{\bar{n}}$ and $A_i \in \mathbb{R}^{l\times\bar{n}}$, for all $i \in \{1, \ldots, B\}$. It is not difficult to check that the DQP problem fits within the template defined by (1)-(2), where

$$f(x) = \sum_{i=1}^{B} \left[\frac{1}{2}\langle x_i, P_i x_i\rangle + \langle x_i, r_i\rangle \right] \quad \text{and} \quad h_i(x_i) = \delta_{\{x\in\mathbb{R}^{\bar{n}}:\|x\|_1\le\omega\}}(x_i), \ \forall i = 1, \ldots, B.$$

We now outline the experimental setup used for the DQP problem. First, to define b, we sample x^b uniformly at random satisfying $\|x_i^b\|_1 \le \omega$ for $i = 1, \ldots, B$ and set $b = \sum_{i=1}^{B} A_i x_i^b$. The initial iteration x^0 is drawn uniformly satisfying $\|x_i^0\|_1 \le \omega$ for $i = 1, \ldots, B$. An orthonormal matrix Q_i is generated using the standard normal distribution. Then, a diagonal matrix D_i is constructed such that one-third of its diagonal entries are set to zero, while the remaining entries are drawn uniformly from the interval $[-10, 10]$, ensuring that at least one of them is negative. Next, the matrix P_i is defined as $P_i = Q_i^\top D_i Q_i$. It is straightforward to verify that if m_i denotes the smallest eigenvalue of D_i, then $m_i < 0$, and the function $f(x_{<i}, \cdot, x_{>i})$ is $|m_i|$-weakly convex. Hence, all variants with constant prox stepsizes set $\lambda_i^0 = 1/(2\max\{1, |m_i|\})$ for $i \in \{1, \ldots, B\}$. Finally, each vector r_i is generated independently, with entries drawn from the standard normal distribution.

The results of the experiments are summarized in Table 1.

Instance			Adapt-ADMM			VP-ADMM		
ω	B	(n, m)	Iters / Mult	Time	$f+h$	Iters / Mult	Time	$f+h$
100	2	(25,25)	**4830**/19	**2.046**	-6.058e+03	+43%/16	+352%	-6.058e+03
		(50,50)	**7474**/19	**4.076**	-7.936e+02	+51%/22	+1038%	-7.936e+02
		(50,75)	**12558**/17	**7.788**	-2.899e+02	+129%/22	+2172%	-3.815e+02
	5	(25,10)	**2598**/30	**1.249**	-2.205e+04	+215%/25	+671%	-2.102e+04
		(25,25)	**5241**/21	**2.744**	-2.126e+04	+48%/22	+280%	-2.108e+04
		(50,50)	**7260**/17	**7.098**	-9.451e+03	+98%/27	+1169%	-9.408e+03
		(50,75)	**15907**/17	**15.287**	-8.303e+03	+102%/15	+1966%	-8.055e+03
	10	(25,10)	**1051**/18	**0.620**	-4.913e+04	+444%/73	+1649%	-4.960e+04
		(25,25)	**2061**/13	**1.623**	-5.257e+04	+749%/14	+3019%	-5.051e+04
		(50,50)	**7078**/12	**10.488**	-2.678e+04	+229%/12	+2766%	-2.715e+04
		(50,75)	**7818**/79	**12.764**	-2.651e+04	+142%/47	+1225%	-2.406e+04
1000	2	(25,25)	**8086**/16	**3.634**	-2.985e+05	+58%/16	+327%	-2.985e+05
		(50,50)	**6948**/14	**4.145**	-1.670e+05	+110%/18	+1420%	-1.717e+05
	5	(25,25)	**12919**/18	**8.411**	-1.993e+06	+55%/38	+308%	-1.972e+06
		(25,50)	**8796**/18	**6.113**	-8.917e+05	+26%/18	+344%	-1.094e+06
		(50,50)	**6407**/20	**5.883**	-1.127e+06	+159%/18	+1508%	-1.257e+06
		(50,75)	+14%/102	**26.804**	-7.187e+05	**20902**/62	+657%	-7.265e+05
	10	(25,10)	**1508**/31	**1.009**	-6.565e+06	+412%/15	+724%	-6.762e+06
		(25,25)	**3270**/33	**2.990**	-4.490e+06	+324%/18	+750%	-4.275e+06
		(25,50)	**6802**/23	**7.481**	-4.101e+06	+181%/45	+993%	-3.531e+06
		(50,50)	**10545**/50	**19.049**	-3.197e+06	+174%/16	+1379%	-3.356e+06
		(50,75)	+229%/20	**191.972**	-3.999e+06	**27809**/25	+126%	-4.039e+06

Table 1: Performance of Adapt-ADMM and VP-ADMM for the DQP problem.

This table does not include the methods Adapt-Penalty, Const-Penalty, Adapt-vADMM, and Const-vADMM, as none of them converged for any of the tested instances. We now make some remarks about the above numerical results. Both methods successfully converged on all instances, but Adapt-ADMM outperformed VP-ADMM on about 90% of the instances tested. In summary, the above results show that Adapt-ADMM is better than its constant stepsize counterpart VP-ADMM.

7.2 Distributed *Cauchy loss* function

This subsection studies the performance of the ADMM variants for finding stationary points of a nonconvex block distributed *Cauchy loss* function problem with B blocks.

For a given pair $(l, \bar{n}) \in \mathbb{N}^2_{++}$, with $l < \bar{n}B$, the B-block *Cauchy loss* function is formulated as

$$\min_{x=(x_1,\ldots,x_B)\in\mathbb{R}^{\bar{n}B}} \left\{ \sum_{i=1}^{B} \frac{\alpha_i^2}{2} \log\left[1 + \left(\frac{y_i - \langle x_i, z_i\rangle}{\alpha_i}\right)^2\right] \;:\; x_i \in \omega\Delta_{\bar{n}},\ \forall i = 1,\ldots,B,\ \text{and}\ \sum_{i=1}^{B} A_i x_i = b \right\},$$

where $\omega > 0$, $b \in \mathbb{R}^l$, $\alpha_i > 0$, $y_i \in \mathbb{R}$, $(z_i, x_i) \in \mathbb{R}^{\bar{n}} \times \mathbb{R}^{\bar{n}}$, and $A_i \in \mathbb{R}^{l\times\bar{n}}$ for all $i \in \{1,\ldots,B\}$. For $m \in \mathbb{R}_{++}$, let $\mathbf{1}_m \in \mathbb{R}^m$ be the vector of all ones. The standard m-dimensional simplex is defined as $\Delta_m = \{x \in \mathbb{R}^m_{++} : \mathbf{1}_m^\top x = 1\}$, and its scaled version is $\omega\Delta_m = \{x \in \mathbb{R}^m_{++} : \mathbf{1}_m^\top x = \omega\}$. It is not difficult to check that the distributed *Cauchy loss* problem fits within the template defined by (1)-(2), where

$$f(x) = \sum_{i=1}^{B} \frac{\alpha_i^2}{2} \log\left[1 + \left(\frac{y_i - \langle x_i, z_i\rangle}{\alpha_i}\right)^2\right] \quad \text{and} \quad h_i(x_i) = \delta_{\omega\Delta_{\bar{n}}}(x_i),\ \forall i = 1,\ldots,B.$$

We now outline the experimental setup used in the above problem. For each $i \in \{1,\ldots,B\}$, the scalar $y_i \in \mathbb{R}$ and the vector $z_i \in \mathbb{R}^{\bar{n}}$ are generated with entries drawn from the standard normal distribution, and the parameter α_i is sampled uniformly at random from the interval $[50, 100]$. To define b, we sample $x^b = (x^b_1,\ldots,x^b_B)$ uniformly at random satisfying $x^b_i \in \omega\Delta_{\bar{n}}$ for $i = 1,\ldots,B$ and set $b = \sum_{i=1}^{B} A_i x^b_i$. The initial iteration $x^0 = (x^0_1,\ldots,x^0_B)$ is drawn uniformly satisfying $x^0_i \in \omega\Delta_{\bar{n}}$ for $i = 1,\ldots,B$.

The results of this experiment are summarized in Table 2. The row labeled "$f + h$" is omitted, as its values consistently ranged from 10^{-12} to 10^{-9} in every instance.

We now present some comments about the numerical results. From these tables, we first observe that the adaptive variants outperform their constant prox stepsize counterparts. Moreover, both Adapt-ADMM and Adapt-Penalty converged on all instances, whereas VP-ADMM and Const-Penalty converged on only 40% of the instances. The standard ADMM performed poorly overall: Adapt-vADMM converged on 36% of the instances, while Const-vADMM converged in only 2% of the cases (one instance). In summary, the tables above show that for the *Cauchy loss* problem, Adapt-ADMM and Adapt-Penalty exhibit a similar behavior.

7.3 Nonconvex QP with Box Constraints

This subsection studies the performance of ADMM variants for solving a class of nonconvex quadratic problems with box constraints (QP-BC).

Specifically, the QP-BC problem considered in this subsection is

$$\min_{x=(x_1,\ldots,x_B)\in\mathbb{R}^{\bar{n}B}} \left\{ \frac{1}{2}\langle x, Px\rangle + \langle r, x\rangle \;:\; \|x\|_\infty \le \omega \text{ and } Ax = b \right\},$$

where $P \in \mathbb{R}^{B\times B}$ is a symmetric indefinite matrix, $A \in \mathbb{R}^{l\times B}$, $(r, b) \in \mathbb{R}^B \times \mathbb{R}^l$, and $\omega > 0$. In this subsection, we view QP-BC as an extreme special case of (1)-(2) where each variable forms a block, and hence $B = n$ and $\bar{n} = 1$. In this case, the variable blocks correspond to individual coordinates. Consequently, each column of the matrix A defines an A_t matrix for $t \in \{1,\ldots,B\}$. It is not difficult to check that the QP-BC problem fits within the template defined by (1)-(2), where

$$f(x) = \frac{1}{2}\langle x, Px\rangle + \langle r, x\rangle \quad \text{and} \quad h_i(x_i) = \delta_{\{x\in\mathbb{R}^{\bar{n}}:\|x\|_\infty\le\omega\}}(x_i),\ \forall i = 1,\ldots,B.$$

We now describe how we orchestrated our QP-BC experiments. First, an orthonormal matrix Q is generated using the standard normal distribution. Then, a diagonal matrix D_i is constructed such that one-third of its

Instance			Adapt-ADMM		VP-ADMM		Adapt-Penalty		Const-Penalty		Adapt-vADMM		Const-vADMM	
ω	B	(n, m)	Iters/Mults	Time	Iters/Mults	Time	Iters	Time	Iters	Time				
100	2	(10,10)	**158** / 1	+18%	+24059% / 1	+107% / 1	+3%	**0.635**	+24541%	+75%	+101%	+41%	*	*
		(25,10)	**130** / 1	**0.093**	+55649% / 1	+2911% / 1	+3%	+5%	+56682%	+2652%	+178%	+159%	*	*
		(25,25)	**712** / 5	+17%	*	*	<1%	**1.639**	*	*	+139%	+123%	*	*
		(50,50)	**755** / 4	+6%	*	*	+3%	**2.075**	*	*	+2718%	+7427%	*	*
		(50,75)	**7194** / 4	**42.344**	*	*	+4%	+6%	*	*	*	*	*	*
		(100,100)	**1801** / 5	<1%	*	*	<1%	**5.148**	*	*	+2351%	+1074%	*	*
		(100,150)	**5449** / 4	+5%	*	*	<1%	**258.139**	*	*	*	*	*	*
	5	(10,10)	**146** / 1	+1642%	+14593% / 3	+6% / 3	+3%	+1494%	+14873%	**1.570**	+9%	+62%	*	*
		(25,10)	**55** / 1	**0.167**	+101656% / 1	+2935% / 1	+4%	+3%	+102953%	+2760%	+47%	+41%	*	*
		(25,25)	**306** / 1	**0.439**	+22699% / 1	+1475% / 1	+2%	<1%	+23135%	+1409%	+276%	+1810%	*	*
		(50,50)	**597** / 1	**1.401**	*	*	+2%	+2%	*	*	+1380%	+1220%	*	*
		(50,75)	**1180** / 1	+1%	*	*	+1%	**59.888**	*	*	+293%	+12%	*	*
		(100,100)	**575** / 1	+2%	*	*	+2%	**13.309**	*	*	+1444%	+146%	*	*
		(100,150)	**551** / 1	**12.601**	*	*	+2%	<1%	*	*	+6456%	+1513%	*	*
	10	(10,10)	**114** / 1	+77%	+16081% / 1	+4% / 1	+3%	+78%	+16340%	**2.863**	+88%	+156%	+25539%	+82%
		(25,10)	+15% / 1	+783%	+22886% / 1	+769% / 1	+17%	+817%	+23265%	+736%	**162**	**0.843**	*	*
		(25,25)	**184** / 1	**3.571**	+30884% / 1	+243% / 1	+3%	<1%	+31374%	+231%	+386%	+328%	*	*
		(50,50)	**431** / 1	+7%	+20886% / 1	+401% / 1	+2%	**6.153**	+21174%	+385%	+690%	+171%	*	*
		(50,75)	**455** / 1	+1%	*	*	+2%	**13.017**	*	*	+366%	+18%	*	*
		(100,100)	**427** / 1	+12%	*	*	+2%	**21.062**	*	*	+1453%	+1825%	*	*
		(100,150)	**507** / 1	+1%	*	*	+2%	**32.618**	*	*	+2473%	+237%	*	*
1000	2	(10,10)	+2% / 5	+2%	*	*	**3377**	**29.445**	*	*	+1478%	+602%	*	*
		(25,10)	**1339** / 1	**1.224**	*	*	<1%	+3%	*	*	+3310%	+4016%	*	*
		(25,25)	<1% / 1	<1%	*	*	**4468**	**2.403**	*	*	*	*	*	*
		(50,50)	<1% / 1	+2%	*	*	**3080**	**6.456**	*	*	*	*	*	*
		(100,100)	<1% / 1	<1%	*	*	**25730**	**70.317**	*	*	*	*	*	*
		(100,150)	**79902** / 1	**513.481**	*	*	+2%	<1%	*	*	*	*	*	*
	5	(10,10)	**405** / 1	+2%	*	*	+2%	**6.591**	*	*	+3527%	+398%	*	*
		(25,10)	**851** / 1	**2.328**	*	*	<1%	+4%	*	*	+2668%	+1325%	*	*
		(25,25)	+2% / 1	+2%	*	*	**476**	**1.164**	*	*	*	*	*	*
		(50,50)	<1% / 1	**4.209**	*	*	**1126**	<1%	*	*	*	*	*	*
		(50,75)	**1990** / 1	**9.221**	*	*	+2%	+3%	*	*	*	*	*	*
		(100,100)	<1% / 1	**20.906**	*	*	**1685**	+2%	*	*	*	*	*	*
		(100,150)	+2% / 1	+2%	*	*	**3647**	**75.915**	*	*	*	*	*	*
	10	(10,10)	**584** / 1	+4%	*	*	<1%	**8.175**	*	*	+727%	+359%	*	*
		(25,10)	<1% / 1	+5%	*	*	**447**	**2.401**	*	*	+4366%	+1524%	*	*
		(25,25)	<1% / 1	**4.535**	*	*	**965**	+11%	*	*	+3532%	+1866%	*	*
		(50,50)	<1% / 1	+5%	*	*	**442**	**5.887**	*	*	*	*	*	*
		(50,75)	<1% / 1	+4%	*	*	**1125**	**9.475**	*	*	*	*	*	*
		(100,100)	**4831** / 1	<1%	*	*	+1%	**104.390**	*	*	*	*	*	*
		(100,150)	**2143** / 1	**65.109**	*	*	<1%	+1%	*	*	*	*	*	*

Bolded values equal to the best algorithm according to iteration count or time. Column "Time" is measured in seconds.

** indicates the algorithm failed to find a stationary point meeting the tolerances by the 100,000th iteration.*

Table 2: Performance of Adapt-ADMM, VP-ADMM, Adapt-Penalty, Const-Penalty, Const-vADMM, and Adapt-vADMM for the Cauchy loss problem.

diagonal entries are set to zero, while the remaining entries are drawn uniformly from the interval $[-10, 10]$, ensuring that at least one of them is negative. Next, the matrix P is defined as $P = Q^\top DQ$. The vector r is generated using the standard normal distribution.

The results of this experiment are summarized in Table 3.

We now make some remarks about the above numerical results. We begin by comparing the performance of the two ADMM variants. Both variants successfully converged on all instances, but Adapt-ADMM outperformed VP-ADMM on about 63% of the instances tested.

We now compare Adapt-ADMM against Adapt-Penalty. While Adapt-ADMM converged successfully in all test instances, Adapt-Penalty converged in only about 63% of them. Moreover, Adapt-ADMM outperformed Adapt-Penalty on about 99% of the test instances.

In summary, the above results confirm the findings of the previous subsections, i.e., Adapt-ADMM is better than its constant stepsize counterpart VP-ADMM, and Adapt-ADMM is more stable than Adapt-Penalty.

8 Concluding Remarks

We start by making some remarks about the analysis of this paper. Even though we have only considered proximal ADMMs, our analysis also applies to proximal penalty methods. If the input $C > 0$ in FP-ADMM

Instance		Adapt-ADMM				VP-ADMM				Adapt-Penalty		
ω	(B,l)	Iters	Time	Mults	$f+h$	Iters	Time	Mults	$f+h$	Iters	Time	$f+h$
1	(50,20)	+74%	+75%	18	-7.280e+01	**2935**	**1.422**	24	-7.309e+01	+1479%	+1468%	-7.309e+01
	(50,40)	**24942**	**15.872**	34	-4.231e+01	+2%	+2%	34	-4.231e+01	+12%	+11%	-4.263e+01
	(100,10)	+197%	+283%	37	-1.418e+02	**641**	**1.407**	17	-1.510e+02	+379%	+604%	-1.431e+02
	(100,25)	**2739**	**8.347**	19	-2.070e+02	+11%	+4%	19	-2.070e+02	*	*	*
	(100,50)	+243%	+245%	101	-1.593e+02	**9113**	**39.417**	12	-1.799e+02	+498%	+476%	-1.522e+02
	(100,75)	**69340**	+1%	20	-5.108e+01	+1%	**373.452**	20	-5.108e+01	*	*	*
10	(50,20)	**2674**	**1.310**	12	-6.293e+03	+668%	+664%	20	-6.528e+03	+89%	+87%	-6.379e+03
	(50,40)	**27770**	**17.865**	13	-5.477e+02	+537%	+535%	23	-6.229e+02	+68%	+67%	-4.488e+02
	(100,10)	**446**	**1.110**	17	-1.504e+04	+178%	+152%	24	-1.519e+04	+3392%	+2975%	-1.463e+04
	(100,25)	+7%	+7%	15	-4.293e+03	**5735**	**16.410**	10	-4.326e+03	*	*	*
	(100,50)	**9170**	**36.270**	21	-6.599e+03	+121%	+121%	16	-6.495e+03	*	*	*
	(100,75)	**42857**	**217.284**	41	-5.032e+03	+187%	+187%	15	-3.907e+03	*	*	*
100	(50,20)	**2530**	**1.236**	18	-8.944e+05	+1%	+1%	17	-9.056e+05	+144%	+141%	-8.944e+05
	(50,40)	**14982**	**9.710**	32	-3.017e+05	+65%	+66%	26	-3.546e+05	+128%	+125%	-3.580e+05
	(100,10)	**2148**	**4.723**	17	-8.557e+05	+123%	+123%	18	-7.527e+05	+12838%	+12789%	-7.892e+05
	(100,25)	**3474**	**9.905**	22	-9.911e+05	+415%	+416%	48	-9.818e+05	+2242%	+2238%	-9.911e+05
	(100,50)	+21%	+21%	45	-8.377e+05	**24711**	**97.683**	23	-9.188e+05	*	*	*
	(100,75)	**19436**	**98.258**	22	-6.714e+05	+168%	+168%	94	-6.770e+05	+739%	+739%	-4.715e+05
1000	(50,20)	**3143**	**1.665**	19	-2.403e+07	+370%	+330%	36	-2.194e+07	*	*	*
	(50,40)	<1%	<1%	73	-1.556e+07	**85465**	**54.416**	75	-1.556e+07	*	*	*
	(100,10)	+95%	+102%	22	-7.742e+07	**1413**	**3.096**	17	-7.958e+07	+25263%	+25244%	-7.742e+07
	(100,25)	**4987**	**14.265**	45	-2.093e+08	+159%	+159%	23	-2.088e+08	*	*	*
	(100,50)	+52%	+53%	42	-3.153e+07	+122%	+122%	20	-3.135e+07	**18548**	**73.107**	-3.266e+07
	(100,75)	**44632**	**226.221**	22	-2.793e+07	+120%	+120%	27	-2.793e+07	*	*	*

Bolded values equal to the best algorithm according to iteration count or time. Column "Time" is measured in seconds.

** indicates the algorithm failed to find a stationary point meeting the tolerances by the 100,000th iteration.*

Table 3: Performance of Adapt-ADMM, VP-ADMM and Adapt-Penalty for the nonconvex QP-BC

is chosen such that $C = \rho$, then FP-ADMM will only perform a single Lagrange multiplier update at its last iteration (see line 7 of FP-ADMM). However, it can be easily seen from our convergence analysis that this last multiplier update is not essential and can be removed, thereby yielding a proximal penalty method that never performs Lagrange multiplier updates. Similarly, if $C = \rho$ in Adapt-ADMM, then a Lagrange multiplier update is performed at of the end of each ℓ-th cycle, i.e., the iterations for which $c = c_{\ell-1}$ (see line 8 of Adapt-ADMM). Since these Lagrange multiplier updates are not essential for our convergence analysis, they can be removed from the description of Adapt-ADMM, resulting in an adaptive proximal penalty method with convergence properties similar to those described in Theorem 6.3.

We now discuss some possible extensions of our analysis in this paper. First, recall that FP-ADMM performs the test in its line 9 to update the Lagrange multiplier, leading to infrequent multiplier updates. It would be interesting to develop proximal ADMM variants with alternative Lagrange multiplier update rules that are more computationally efficient than the one used by FP-ADMM. Second, it would be interesting to develop proximal ADMM variants for composite optimization problems with block constraints given by $\sum_{t=1}^{B} g_t(x_t) \le 0$ where the components of $g_t : \mathbb{R}^{n_t} \to \mathbb{R}^l$ are convex for each $t = 1, \ldots, B$. Third, this paper assumes that $\operatorname{dom} h$ is bounded (see assumption (A1)) and h restricted to its domain is Lipschitz continuous (see assumption (A1)). It would be interesting to extend its analysis to the case where these assumptions are removed. Finally, our analysis shows that $(\hat{x}, \hat{p}, \hat{v}, \hat{\varepsilon}, \hat{c})$ output by Adapt-ADMM satisfies $\|A\hat{x} - b\| \le \eta$ whenever $\hat{c} = \mathcal{O}(1/\eta)$. It would be interesting to investigate ADMM variants that guarantees this same η-feasibility under weaker conditions on c, e.g., $c = \mathcal{O}(1)$. Some efforts along this direction have been made in [64, 65] under more restrictive conditions on problem (1) than the ones assumed in this paper.

A Technical Results for Proof of Lagrange Multipliers

This appendix provides some technical results to show that under certain conditions the sequence of Lagrange multipliers generated by FP-ADMM is bounded.

The next two results, used to prove Lemma A.3, can be found in [19, Lemma B.3] and [35, Lemma 3.10], respectively.

Lemma A.1 *Let $A : \mathbb{R}^n \to \mathbb{R}^l$ be a nonzero linear operator. Then,*

$$\nu_A^+ \|u\| \le \|A^* u\|, \quad \forall u \in A(\mathbb{R}^n).$$

Lemma A.2 *Let h be a function as in (A1). Then, for every $\delta \ge 0$, $z \in \mathcal{H}$, and $\xi \in \partial_\delta h(z)$, we have*

$$\|\xi\| \mathrm{dist}(u, \partial \mathcal{H}) \le [\mathrm{dist}(u, \partial\mathcal{H}) + \|z-u\|]\, M_h + \langle \xi\, , z-u\rangle + \delta \quad \forall u \in \mathcal{H}$$

where $\partial\mathcal{H}$ denotes the boundary of $\mathcal{H}$.

The following result, whose statement is in terms of the δ-subdifferential instead of the classical subdifferential, is a slight generalization of [58, Lemma B.3]. For the sake of completeness, we also include its proof.

Lemma A.3 *Assume that $b \in \mathbb{R}^l$, linear operator $A : \mathbb{R}^n \to \mathbb{R}^l$ and function $h(\cdot)$, satisfy assumptions (A4) and (A1), respectively. If $(q^-, \varrho) \in A(\mathbb{R}^n) \times (0, \infty)$ and $(z, q, r, \delta) \in \mathrm{dom}\, h \times \mathbb{R}^l \times \mathbb{R}^n \times \mathbb{R}_+$ satisfy*

$$q = q^- + \varrho(Az - b) \quad \textit{and} \quad r \in \partial_\delta h(z) + A^* q, \tag{76}$$

then we have

$$\|q\| \le \max\left\{\|q^-\|,\ \Xi\Big(\|z-\bar{x}\|\ ,\ \|r\| + \delta\Big)\right\} \tag{77}$$

where $\bar{\mathrm{x}}$ is as in (A4),

$$\Xi(s,t) := \frac{t + \left(s + \bar{d}\right)(M_h + t)}{\bar{d}\nu_A^+} \quad \forall (s,t) \in \mathbb{R}^2_+, \tag{78}$$

M_h and $\bar{d} > 0$ are as in (A1) and (A4), respectively, and ν_A^+ is the smallest positive singular value of A.

Proof: We first claim that

$$\bar{d}\nu_A^+ \|q\| \le (\|z - \bar{x}\| + \bar{d})\,(M_h + \|r\|) - \langle q\, , Az - b\rangle + \delta \tag{79}$$

holds. The assumption on (z, q, r, δ) implies that $r - A^*q \in \partial_\delta h(z)$. Hence, using the Cauchy-Schwarz inequality, the definitions of $\bar{d}$ and $\bar{x}$ in (A4), and Lemma A.2 with $\xi = r - A^*q$, and $u = \bar{x}$, we have:

$$\bar{d}\|r - A^*q\| - \left[\bar{d} + \|z-\bar{x}\|\right] M_h \overset{(A.2)}{\le} \langle r - A^*q\, , z - \bar{x}\rangle + \delta \le \|r\|\|z-\bar{x}\| - \langle q\, , Az-b\rangle + \delta. \tag{80}$$

Now, using the above inequality and the triangle inequality, we conclude that:

$$\bar{d}\|A^*q\| + \langle q\, , Az - b\rangle \overset{(80)}{\le} \left[\bar{d} + \|z-\bar{x}\|\right] M_h + \|r\|\left(\|z-\bar{x}\| + \bar{d}\right) + \delta = (\|z-\bar{x}\| + \bar{d})\,(M_h + \|r\|) + \delta. \tag{81}$$

We note that $q \in A(\mathbb{R}^n)$ follows immediately from the identity in (77), the hypothesis that $q^- \in A(\mathbb{R}^n)$, and the fact that $b \in \mathrm{Im}(A)$ due to assumption (A4). Hence, inequality (79) now follows from the above inequality and Lemma A.1.

We now prove (77). Relation (76) implies that $\langle q\, , Az - b\rangle = \|q\|^2/\varrho - \langle q^-\, , q\rangle/\varrho$, and hence that

$$\bar{d}\nu_A^+\|q\| + \frac{\|q\|^2}{\varrho} \le (\|z-\bar{x}\| + \bar{d})(M_h + \|r\|) + \frac{\langle q^-\, , q\rangle}{\varrho} + \delta \le (\|z-\bar{x}\| + \bar{d})(M_h + \|r\|) + \frac{\|q\|}{\varrho}\|q^-\| + \delta, \tag{82}$$

where the last inequality is due to the Cauchy-Schwarz inequality. Now, letting W denote the right hand side of (77) and using (82), we conclude that

$$\begin{aligned}\left(\bar{d}\nu_A^+ + \frac{\|q\|}{\varrho}\right)\|q\| &\overset{(82)}{\le} \left(\frac{(\|z-\bar{x}\| + \bar{d})(M_h + \|r\|) + \delta}{W} + \frac{\|q\|}{\varrho}\right) W \\ &\le \left(\frac{(\|z-\bar{x}\| + \bar{d})M_h + (\|z-\bar{x}\| + \bar{d} + 1)(\|r\| + \delta)}{W} + \frac{\|q\|}{\varrho}\right) W \le \left(\bar{d}\nu_A^+ + \frac{\|q\|}{\varrho}\right) W,\end{aligned} \tag{83}$$

and hence that (77) holds. ∎

We conclude this section with a technical result of convexity which is used in the proof of Lemma 6.4. Its proof can be found in [44, Lemma A1] but for the sake of completeness a more detailed proof is given here.

Lemma A.4 *Assume that $\xi > 0$, $\psi \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$ and positive definite real-valued $n \times n$-matrix Q are such that $\psi - (\xi/2)\|\cdot\|_Q^2$ is convex and let $(y, v, \eta) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+$ be such that $v \in \partial_\eta \psi(y)$. Then, for any $\tau > 0$,*

$$\psi(u) \geq \psi(y) + \langle v\,, u - y\rangle - (1 + \tau^{-1})\eta + \frac{(1+\tau)^{-1}\xi}{2}\|u - y\|_Q^2 \quad \forall u \in \mathbb{R}^n. \tag{84}$$

Proof: Let $\psi_v := \psi - \langle v\,, \cdot\rangle$. The assumptions imply that ψ_v has a unique global minimum $\bar{y}$ and that

$$\psi_v(u) \geq \psi_v(\bar{y}) + \frac{\xi}{2}\|u - \bar{y}\|_Q^2 \geq \psi_v(y) - \eta + \frac{\xi}{2}\|u - \bar{y}\|_Q^2 \tag{85}$$

for every $u \in \mathbb{R}^n$. The above inequalities with $u = y$ imply that $(\xi/2)\|\bar{y} - y\|_Q^2 \leq \eta$. On the other hand, for any $\tilde{u}, u' \in \mathbb{R}^n$ and $\tau > 0$, it holds

$$\begin{aligned}\|\tilde{u} + u'\|^2 &= \|\tilde{u}\|^2 + \|u'\|^2 + 2\left\langle \frac{1}{\sqrt{\tau}}\tilde{u}\,, \sqrt{\tau}u'\right\rangle \leq \|\tilde{u}\|^2 + \|u'\|^2 + \frac{1}{\tau}\|\tilde{u}\|^2 + \tau\|u'\|^2\\ &= (1+\tau)\|u'\|^2 + (1+\tau^{-1})\|\tilde{u}\|^2\end{aligned}$$

which implies in

$$(1+\tau)^{-1}\|\tilde{u} + u'\|^2 \leq \|u'\|^2 + (1+\tau)^{-1}(1+\tau^{-1})\|\tilde{u}\|^2 = \|u'\|^2 + \tau^{-1}\|\tilde{u}\|^2.$$

Hence, adding and subtracting the term $(\tau^{-1}\xi/2)\|\bar{y} - y\|_Q^2$ in the right hand side of (85) and using the previous inequality with $\tilde{u} = u - \bar{y}$ and $u' = \bar{y} - y$, we obtain that

$$\begin{aligned}\psi_v(u) &\geq \psi_v(y) - \eta - \frac{\tau^{-1}\xi}{2}\|\bar{y} - y\|_Q^2 + \frac{\xi}{2}\left(\tau^{-1}\|y - \bar{y}\|_Q^2 + \|u - \bar{y}\|_Q^2\right)\\ &\geq \psi_v(y) - (1+\tau^{-1})\eta + \frac{(1+\tau)^{-1}\xi}{2}\|u - y\|_Q^2\end{aligned}$$

for every $u \in \mathbb{R}^n$. Hence, (84) follows from the above conclusion and the definition of ψ_v. ∎

B ADAP-FISTA algorithm

This appendix section presents an adaptive variant of ACG, called ADAP-FISTA, for solving (15) under the assumption that (B1), (B2), and $\nabla\psi_{\mathrm{s}}(\cdot)$ is $\tilde{M}$-Lipschitz continuous, i.e.,

$$\|\nabla\psi_{\mathrm{s}}(z') - \nabla\psi_{\mathrm{s}}(z)\| \leq \tilde{M}\|z' - z\| \quad \forall z, z' \in \mathbb{R}^n. \tag{86}$$

We would like to emphasize that the notations introduced in this appendix, related to the ADAP-FISTA, are local to this section and should not be confused with those used in previous sections. These choices are made to remain consistent with the original presentation of the algorithm in [58, Appendix A], and they do not carry the same interpretation as in the rest of the paper.

Before formally stating ADAP-FISTA, we give some comments. ADAP-FISTA requires as input an arbitrary positive estimate M_0 for the unknown parameter $\tilde{M}$. Moreover, ADAP-FISTA is a variant of SFISTA [3, 4, 51], which in turn is an ACG variant that solves instances of (15) with ψ_{s} strongly convex and that requires the availability of a strong convex parameter for ψ_{s}. Since ADAP-FISTA is an enhanced version of SFISTA, it also uses as input a good guess μ_0 for what is believed to be a strong convex parameter of ψ_{s} (even though such parameter may not exist as ψ_{s} is not assumed to be strongly convex). In other words, ADAP-FISTA is used under the belief that ψ_{s} is μ_0-strongly convex. If a key test inequality within ADAP-FISTA fails to be satisfied then it stops without finding a $(\sqrt{\sigma}; x_0)$-relative stationary solution of (15), but reaches the important conclusion that ψ_{s} is not μ_0-strongly convex.

We are now ready to present the ADAP-FISTA algorithm below.

ADAP-FISTA Method

Input: $(x_0, M_0, \mu_0, \sigma) \in \mathrm{dom}\,\psi_{\mathrm{n}} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that $M_0 > \mu_0$.

0. Let $\chi \in (0,1)$ and $\beta > 1$ be given, and set $y_0 = x_0$, $A_0 = 0$, $\tau_0 = 1$, and $j = 0$.

1. Set $M_{j+1} = M_j$.

2. Compute

$$a_j = \frac{\tau_j + \sqrt{\tau_j^2 + 4\tau_j A_j (M_{j+1} - \mu_0)}}{2(M_{j+1} - \mu_0)}, \quad \tilde{x}_j = \frac{A_j y_j + a_j x_j}{A_j + a_j},$$

$$y_{j+1} := \underset{v \in \operatorname{dom} \psi_{\rm n}}{\operatorname{argmin}} \left\{ q(v; \tilde{x}_j, M_{j+1}) := \psi_{\rm s}(\tilde{x}_j) + \langle \nabla \psi_{\rm s}(\tilde{x}_j), v - \tilde{x}_j \rangle + \psi_{\rm n}(v) + \frac{M_{j+1}}{2} \|v - \tilde{x}_j\|^2 \right\}. \tag{87}$$

If the inequality

$$\psi_{\rm s}(\tilde{x}_j) + \langle \nabla \psi_{\rm s}(\tilde{x}_j), y_{j+1} - \tilde{x}_j \rangle + \frac{(1-\chi) M_{j+1}}{2} \|y_{j+1} - \tilde{x}_j\|^2 \geq \psi_{\rm s}(y_{j+1}) \tag{88}$$

holds, then go to step 3; else, set $M_{j+1} \leftarrow \beta M_{j+1}$ and repeat step 2.

3. Compute

$$\begin{aligned} A_{j+1} &= A_j + a_j, \quad \tau_{j+1} = \tau_j + a_j \mu_0, \\ s_{j+1} &= (M_{j+1} - \mu_0)(\tilde{x}_j - y_{j+1}), \\ x_{j+1} &= \frac{1}{\tau_{j+1}} \left[\mu_0 a_j y_{j+1} + \tau_j x_j - a_j s_{j+1} \right]. \end{aligned}$$

4. If the inequality

$$\|y_{j+1} - x_0\|^2 \geq \chi A_{j+1} M_{j+1} \|y_{j+1} - \tilde{x}_j\|^2, \tag{89}$$

holds, then go to step 5; otherwise, stop with **failure**.

5. Compute

$$u_{j+1} = \nabla \psi_{\rm s}(y_{j+1}) - \nabla \psi_{\rm s}(\tilde{x}_j) + M_{j+1}(\tilde{x}_j - y_{j+1}).$$

If the inequality

$$\|u_{j+1}\| \leq \sqrt{\sigma} \|y_{j+1} - x_0\| \tag{90}$$

holds, then stop with **success** and output $(y, u) := (y_{j+1}, u_{j+1})$; otherwise, $j \leftarrow j+1$ and go to step 1.

We now make some remarks about ADAP-FISTA. First, steps 2 and 3 of ADAP-FISTA appear in the usual SFISTA for solving strongly convex version of (15), either with a static Lipschitz constant (i.e., $M_{j+1} = \tilde{M}$ for all $j \geq 0$), or with adaptive line search for M_{j+1} (e.g., as in step 2 of ADAP-FISTA). Second, the pair (y_{j+1}, u_{j+1}) always satisfies the inclusion in (18) (see [58, Lemma A.3]); hence, if ADAP-FISTA stops successfully in step 5, then the triple $(y_{j+1}, u_{j+1}, 0)$ is a (σ, x_0)-relative stationary solution of (15), due to (90). Finally, if condition (89) in step 4 is never violated, then ADAP-FISTA must stop successfully in step 5 (see Proposition B.1 below).

The following result describes the main properties of ADAP-FISTA.

Proposition B.1 *Assume that (B1) and (B2) hold and that $\nabla \psi_{\rm s}(\cdot)$ is $\tilde{M}$-Lipschitz continuous. Then, the following statements about the* ADAP-FISTA *method with arbitrary input $(x_0, M_0, \mu_0, \sigma) \in \operatorname{dom} \psi_{\rm n} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++}$ hold:*

(a) it always stops (with either success or failure) in at most

$$\mathcal{O}\left(\sqrt{\frac{\tilde{M} + M_0}{\mu_0}} \max\left\{ \log_2(\sigma^{-1/2} \tilde{M}), 1 \right\} \right) \tag{91}$$

iterations/resolvent evaluations;

(b) if it stops successfully with output (y,u), then the triple $(y,u,0)$ is a $(\sqrt{\sigma};x_0)$-relative stationary solution of (15) *(see Definition 2.1);*

(c) if $\psi_{\rm s}(\cdot)$ is μ_0-strongly convex, then ADAP-FISTA *always terminates successfully, and therefore with a $(\sqrt{\sigma};x_0)$-relative stationary solution of* (15).

We now make some remarks about Proposition B.1. First, if ADAP-FISTA fails then it follows from Proposition B.1(c) that $\psi_{\rm s}$ is not $\tilde{\mu}$-strongly convex. Hence, failure of the method sends the message that $\psi_{\rm s}$ is not "desirable", i.e., is far from being $\tilde{\mu}$-strongly convex. Second, if ADAP-FISTA successfully terminates (which can happen even if $\psi_{\rm s}$ is not $\tilde{\mu}$-strongly convex), then Proposition B.1(b) guarantees that it finds the desired relative stationary solution. Third, if $\sigma^{-1/2} = \mathcal{O}(1)$ and $M_0 = \mathcal{O}(\tilde{M})$, then (91) reduces to $\mathcal{O}((\tilde{M}/\mu_0)^{1/2})$.

C Inexact Solution Concept

This section shows how near-stationary solutions, which are absolute analogues of the ones considered in Definition 2.1, yield points with nearly nonnegative directional derivatives along all unit directions. For a given function ϕ and $y \in \operatorname{dom}\phi$ such that the directional derivative $\phi'(y;u-y)$ is well-defined for every $u \in \mathbb{R}^n$, define

$$\Theta(y;\phi) := -\inf_{u\in\mathbb{R}^n}\{\phi'(y;u-y) \,:\, \|u-y\|\le 1\}. \tag{92}$$

Clearly, $\Theta(y;\phi)\ge 0$ and equality holds if and only if $\phi'(y;u-y)\ge 0$ for every $u\in\mathbb{R}^n$. If $\phi\in\overline{\operatorname{Conv}}\,(\mathbb{R}^n)$, the latter condition is equivalent to y being an optimal solution of ϕ, or equivalently, the inclusion $0\in\partial\phi(y)$. More generally, a point y such that $\Theta(y;\phi)=0$ is referred to as a stationary point of ϕ.

More generally, y is called a directional near-stationary point when $\Theta(y;\phi)$ is near zero. This section discusses how an absolute analogue of the stationary condition in Definition 2.1 yields directional near-stationary points and related ones expressed in terms of subdifferentials.

The main result of this section is Proposition C.3, stated in the setting of a nonconvex composite optimization problem. Its proof requires two preliminary technical lemmas.

Lemma C.1 *Let $\lambda>0$, $\varepsilon\ge 0$, function $\phi\in\overline{\operatorname{Conv}}\,(\mathbb{R}^n)$, and $x\in\operatorname{dom}\phi$ such that $0\in\partial_\varepsilon\phi(x)$ be given, and let y denote the unique optimal solution of the strongly convex optimization problem*

$$y := \arg\min_{u\in\mathbb{R}^n}\left\{\phi_\lambda(u) := \phi(u) + \frac{1}{2\lambda}\|u-x\|^2\right\}. \tag{93}$$

Then, $y\in\mathbb{R}^n$ satisfies $\|y-x\|\le\sqrt{\varepsilon\lambda}$.

Proof: Let $\lambda>0$, $\varepsilon\ge 0$, and $x\in\operatorname{dom}\phi$ such that $0\in\partial_\varepsilon\phi(x)$ be given. Using the definition of y and the fact that ϕ_λ is $(1/\lambda)$-strongly convex, we have $\phi_\lambda(u)\ge\phi_\lambda(y)+\|u-y\|^2/(2\lambda)$ for every $u\in\mathbb{R}^n$, or equivalently,

$$\phi(u)+\frac{1}{2\lambda}\|u-x\|^2 \ge \phi(y)+\frac{1}{2\lambda}\|y-x\|^2+\frac{1}{2\lambda}\|u-y\|^2 \quad \forall u\in\mathbb{R}^n.$$

The above inequality with $u=x$ implies that $\|y-x\|^2/\lambda\le\phi(x)-\phi(y)\le\varepsilon$, where the last inequality is due to the assumption that $0\in\partial_\varepsilon\phi(x)$ and the definition of the ε-subdifferential in (10). Thus, the conclusion of the lemma holds. ∎

The proof of the next well-known result can be found for example in [28, Lemma F.1.2] (see also Chapter 8 of [56] and [8]).

Lemma C.2 *Let $\psi_{\rm n}\in\overline{\operatorname{Conv}}\,(\mathbb{R}^n)$, (possibly nonconvex) differentiable function $\psi_{\rm s}$ on $\operatorname{dom}\psi_{\rm n}$, and $(y,w)\in\operatorname{dom}\psi_{\rm n}\times\mathbb{R}^n$ such that $w\in\nabla\psi_{\rm s}(y)+\partial\psi_{\rm n}(y)$ be given. Then,*

$$\Theta(y;f+h) = \operatorname{dist}\left(0,\nabla\psi_{\rm s}(y)+\partial\psi_{\rm n}(y)\right)\le\|w\|,$$

where $\Theta(y;f+h)$ is as in (92).

We are now ready to state the main result of this subsection.

Proposition C.3 *Let $\psi_{\rm n} \in \overline{\rm Conv}\,(\mathbb{R}^n)$, (possibly nonconvex) differentiable function $\psi_{\rm s}$ on $\operatorname{dom}\psi_{\rm n}$, and $(x,r,\varepsilon) \in \operatorname{dom}\psi_{\rm n} \times \mathbb{R}^n \times \mathbb{R}_+$ such that*

$$r \in \nabla\psi_{\rm s}(x) + \partial_\varepsilon \psi_{\rm n}(x), \tag{94}$$

be given, and define

$$y = y(x,r) := \arg\min_{u\in\mathbb{R}^n} \left\{ \langle \nabla\psi_{\rm s}(x) - r, u\rangle + \psi_{\rm n}(u) + \frac{1}{2}\|u-x\|^2 \right\}, \tag{95}$$

$$w = w(x,r) := x - y + r + [\nabla\psi_{\rm s}(y) - \nabla\psi_{\rm s}(x)]. \tag{96}$$

Then, the following statements hold:

a) $(y,w) = (y(x,r), w(x,r)) \in \operatorname{dom}\psi_{\rm n} \times \mathbb{R}^n$ *satisfies*

$$w \in \nabla\psi_{\rm s}(y) + \partial\psi_{\rm n}(y), \qquad \|y-x\| \le \sqrt{\varepsilon}, \tag{97}$$

and the inequality

$$\|w\| \le \|\nabla\psi_{\rm s}(y) - \nabla\psi_{\rm s}(x)\| + \sqrt{\varepsilon} + \|r\|. \tag{98}$$

b) if, in addition, $\operatorname{dom}\psi_{\rm n}$ is compact and $\nabla\psi_{\rm s}$ is continuous on $\operatorname{dom}\psi_{\rm n}$, then for any $\eta > 0$, there exists $\delta > 0$ satisfying the following property: the pair $(y,w) = (y(x,r), w(x,r))$ associated with any (x,r,ε) such that inclusion (94) and the inequality $\|r\|^2 + 2\varepsilon \le \delta^2/2$ holds, satisfies

$$\|y-x\| \le \eta, \qquad \|w\| \le \eta; \tag{99}$$

as a consequence,

$$\Theta(y;\psi_{\rm s}+\psi_{\rm n}) = \operatorname{dist}\left(0, \nabla\psi_{\rm s}(y) + \partial\psi_{\rm n}(y)\right) \le \eta. \tag{100}$$

Proof: (a) The inclusion in (97) follows from the fact that y satisfies the optimality condition for (95) and the definition of w in (96). Now, define

$$\phi(\cdot) := \langle \nabla\psi_{\rm s}(x) - r, \cdot\rangle + \psi_{\rm n}(\cdot)$$

and note that inclusion in (94) implies that $0 \in \partial_\varepsilon\phi(x)$. Proposition C.1 with $\lambda = 1$ and the definition of y in (95) then imply that the inequality in (97) holds. Now, the inclusion in (97) and Lemma C.2 imply that the first inequality in (98) holds. Finally, the definition of w, the triangular inequality, and the inequality in (97), imply that the second inequality in (98) also holds. We have proved that (a) holds.

(b) Let $\eta > 0$ be given. The additional assumptions made on this statement imply that $\nabla\psi_{\rm s}$ is uniformly continuous on $\operatorname{dom}\psi_{\rm n}$. Thus, there exists $\rho > 0$ such that

$$\|z-x\| \le \rho \Rightarrow \|\nabla\psi_{\rm s}(x) - \nabla\psi_{\rm s}(z)\| \le \frac{\eta}{2}. \tag{101}$$

We now show that the scalar $\delta := \min\{2\rho, \eta/2\}$ fulfills the conclusion of this statement. Indeed, let triple (x,r,ε) satisfying inclusion (94) and the inequality $\|r\|^2 + 2\varepsilon \le \delta^2/2$ be given. Clearly, the last inequality, together with the definition of δ and the inequality in (97), implies that

$$\|r\| + \sqrt{\varepsilon} \le \left(2\|r\|^2 + 2\varepsilon\right)^{1/2} \le \delta \le \frac{\eta}{2}, \qquad \|y-x\| \le \sqrt{\varepsilon} \le \sqrt{\frac{\delta^2}{4}} = \frac{\delta}{2} \le \min\{\rho,\eta\}$$

These two inequalities, the inequalities in (98), and implication (101) with $z = y$, then show that (99) holds. Results in (100) follow by Lemma C.2 and the last inequality in (99). ∎

Note that if the function $\psi_{\rm s}$ in Proposition C.3 is also L-smooth on $\operatorname{dom}\psi_{\rm n}$, then it follows from (98) that

$$\|w\| \le \sqrt{\varepsilon}(L+1) + \|r\|.$$

Hence, without the compactness assumption on $\mathrm{dom}\,\psi_\mathrm{n}$, it can be easily seen that the conclusion of statement (b) holds if δ is chosen as $\delta := \eta/[2(L+1)]$.

Finally, the usual subdifferential for convex functions has been generalized to functions of the form $\psi_\mathrm{s} + \psi_\mathrm{n}$ where $(\psi_\mathrm{s}, \psi_\mathrm{n})$ are as in Lemma C.2 or Proposition C.3 (e.g., see [55, Ch. 10], [47]). This more general subdifferential, denoted by $\tilde{\partial}(\psi_\mathrm{s} + \psi_\mathrm{n})$, has the property that

$$\tilde{\partial}(\psi_\mathrm{s} + \psi_\mathrm{n})(y) = \nabla\psi_\mathrm{s}(y) + \partial\psi_\mathrm{n}(y) \quad \forall y \in \mathrm{dom}\,\psi_\mathrm{n}.$$

Hence, the above inequalities involving the quantity $\mathrm{dist}\,(0, \nabla\psi_\mathrm{s}(y) + \partial\psi_\mathrm{n}(y))$ can equivalently be rewritten in terms of $\mathrm{dist}(0, \tilde{\partial}(\psi_\mathrm{s} + \psi_\mathrm{n})(y))$.

References

[1] N. Aybat and G. Iyengar. A first-order smoothed penalty method for compressed sensing. *SIAM J. Optim.*, 21(1):287–313, 2011.

[2] N. Aybat and G. Iyengar. A first-order augmented Lagrangian method for compressed sensing. *SIAM J. Optim.*, 22(2):429–459, 2012.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[4] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009.

[5] D. P. Bertsekas. *Nonlinear programming*. Taylor & Francis, 3ed edition, 2016.

[6] E. Birgin, G. Haeser, and J. M. Martínez. Safeguarded augmented Lagrangian algorithms with scaled stopping criterion for the subproblems. *Computational Optimization and Applications*, 91:491–509, 2025.

[7] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[8] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25(5):1197–1211, 1988.

[9] M. T. Chao, Y. Zhang, and J. B. Jian. An inertial proximal alternating direction method of multipliers for nonconvex optimization. *International Journal of Computer Mathematics*, 98(6):1199–1217, 2021.

[10] J. Eckstein and D. P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

[11] J. Eckstein and M. C. Ferris. Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control. *INFORMS Journal on Computing*, 10(2):218–235, 1998.

[12] J. Eckstein and M. Fukushima. Some reformulations and applications of the alternating direction method of multipliers. In *Large scale optimization*, pages 115–134. Springer, 1994.

[13] J. Eckstein and B. F. Svaiter. A family of projective splitting methods for the sum of two maximal monotone operators. *Mathematical Programming*, 111(1):173–199, 2008.

[14] J. Eckstein and B. F. Svaiter. General projective splitting methods for sums of maximal monotone operators. *SIAM Journal on Control and Optimization*, 48(2):787–811, 2009.

[15] M. I. Florea and S. A. Vorobyov. An accelerated composite gradient method for large-scale composite objective problems. *IEEE Transactions on Signal Processing*, 67(2):444–459, 2018.

[16] D. Gabay. Applications of the method of multipliers to variational inequalities. In *Studies in mathematics and its applications*, volume 15, pages 299–331. Elsevier, 1983.

[17] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976.

[18] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[19] M. L. N. Goncalves, J. G. Melo, and R. D. C. Monteiro. Convergence rate bounds for a proximal ADMM with over-relaxation stepsize parameter for solving nonconvex linearly constrained problems. *Pacific Journal of Optimization*, 15:379–398, 2019.

[20] D. Hajinezhad and M. Hong. Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization. *Math. Program.*, 176:207–245, 2019.

[21] Y. He and R. Monteiro. An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM J. Optim.*, 26(1):29–56, 2016.

[22] Y. He and R. D. C. Monteiro. Accelerating block-decomposition first-order methods for solving composite saddle-point and two-player Nash equilibrium problems. *SIAM J. Optim.*, 25:2182–2211, 2015.

[23] J. B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms II. Advanced Theory and Bundle Methods*. Springer, Berlin, 1993.

[24] M. Hong, Z.-Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[25] A. Izmailov, M. Solodov, and E. Uskov. Global convergence of augmented Lagrangian methods applied to optimization problems with degenerate constraints, including problems with complementarity constraints. *SIAM Journal on Optimization*, 22:1579–1606, 2012.

[26] Z. Jia, J. Huang, and Z. Wu. An incremental aggregated proximal ADMM for linearly constrained nonconvex optimization with application to sparse logistic regression problems. *Journal of Computational and Applied Mathematics*, 390:113384, 2021.

[27] B. Jiang, T. Lin, S. Ma, and S. Zhang. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. *Computational Optimization and Applications*, 72(1):115–157, 2019.

[28] W. Kong. Accelerated inexact first-order methods for solving nonconvex composite optimization problems, 2021. URL https://arxiv.org/abs/2104.09685.

[29] W. Kong. Complexity-optimal and parameter-free first-order methods for finding stationary points of composite optimization problems. *SIAM Journal on Optimization*, 34(3):3005–3032, 2024.

[30] W. Kong and R. D. C. Monteiro. An accelerated inexact dampened augmented Lagrangian method for linearly constrained nonconvex composite optimization problems. *Comput. Optim. Appl.*, 2023.

[31] W. Kong and R. D. C. Monteiro. Global complexity bound of a proximal ADMM for linearly constrained nonseparable nonconvex composite programming. *SIAM Journal on Optimization*, 34(1):201–224, 2024.

[32] W. Kong, J. G. Melo, and R. D. Monteiro. An efficient adaptive accelerated inexact proximal point method for solving linearly constrained nonconvex composite problems. *Computational Optimization and Applications*, 76:305–346, 2020.

[33] W. Kong, J. G. Melo, and R. Monteiro. FISTA and Extensions - Review and New Insights. *Optimization Online*, 2021.

[34] W. Kong, J. G. Melo, and R. D. Monteiro. Iteration complexity of a proximal augmented Lagrangian method for solving nonconvex composite optimization problems with nonlinear convex constraints. *Mathematics of Operations Research*, 48(2):1066–1094, 2023.

[35] W. Kong, J. G. Melo, and R. D. C. Monteiro. Iteration complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical Lagrangian function. *SIAM Journal on Optimization*, 33(1):181–210, 2023.

[36] G. Lan and R. D. C. Monteiro. Iteration-complexity of first-order penalty methods for convex programming. *Math. Program.*, 138(1):115–139, 2013.

[37] G. Lan and R. D. C. Monteiro. Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Math. Program.*, 155(1):511–547, 2016.

[38] Y. Liu, X. Liu, and S. Ma. On the nonergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming. *Math. Oper. Res.*, 44(2):632–650, 2019.

[39] P.-L. Loh. Statistical consistency and asymptotic normality for high-dimensional robust M-estimators. *The Annals of Statistics*, 45(2):866 – 896, 2017. doi: 10.1214/16-AOS1471. URL https://doi.org/10.1214/16-AOS1471.

[40] Z. Lu and Z. Zhou. Iteration-complexity of first-order augmented Lagrangian methods for convex conic programming. *SIAM journal on optimization*, 33(2):1159–1190, 2023.

[41] J. Melo, R. D. C. Monteiro, and H. Wang. Iteration-complexity of an inexact proximal accelerated augmented Lagrangian method for solving linearly constrained smooth nonconvex composite optimization problems. *Available on arXiv:2006.08048*, 2020.

[42] J. G. Melo and R. D. C. Monteiro. Iteration-complexity of a linearized proximal multiblock ADMM class for linearly constrained nonconvex optimization problems. *Optimization Online preprint*, 2017.

[43] J. G. Melo and R. D. C. Monteiro. Iteration-complexity of a Jacobi-type non-Euclidean ADMM for multi-block linearly constrained nonconvex programs. *Available on arXiv:1705.07229*, 2017.

[44] J. G. Melo, R. D. Monteiro, and H. Wang. A proximal augmented Lagrangian method for linearly constrained nonconvex composite optimization problems. *Journal of Optimization Theory and Applications*, 202(1):388–420, 2024.

[45] R. Monteiro, Ortiz, and B. F. Svaiter. An adaptive accelerated first-order method for convex optimization. *Comput. Optim. Appl.*, 64:31–73, 2016.

[46] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23(1):475–507, 2013.

[47] B. Mordukhovich. Variational analysis and generalized differentiation, II: Applications, 2006.

[48] I. Necoara, A. Patrascu, and F. Glineur. Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2):305–335, 2019.

[49] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1): 125–161, 2013.

[50] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O\left(\frac{1}{k^2}\right)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[51] Y. E. Nesterov. *Introductory lectures on convex optimization : a basic course.* Kluwer Academic Publ., 2004.

[52] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst for gradient-based nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 613–622. PMLR, 2018.

[53] A. Patrascu, I. Necoara, and Q. Tran-Dinh. Adaptive inexact fast augmented Lagrangian methods for constrained convex optimization. *Optim. Lett.*, 11(3):609–626, 2017.

[54] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

[55] R. T. Rockafellar and R. J. Wets. *Variational analysis*. Springer, 1998.

[56] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer Science & Business Media, Berlin, 2009.

[57] A. Ruszczyński. An augmented Lagrangian decomposition method for block diagonal linear programming problems. *Operations Research Letters*, 8(5):287–294, 1989.

[58] A. Sujanani and R. D. C. Monteiro. An adaptive superfast inexact proximal augmented Lagrangian method for smooth nonconvex composite optimization problems. *Journal of Scientific Computing*, 97(2): 34, 2023.

[59] K. Sun and X. A. Sun. Dual descent augmented Lagrangian method and alternating direction method of multipliers. *SIAM Journal on Optimization*, 34(2):1679–1707, 2024.

[60] A. Themelis and P. Patrinos. Douglas–Rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM Journal on Optimization*, 30(1):149–181, 2020.

[61] Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

[62] Y. Xu. Iteration complexity of inexact augmented Lagrangian methods for constrained convex programming. *Mathematical Programming*, 185:199–244, 2021.

[63] J. Zeng, W. Yin, and D. Zhou. Moreau envelope augmented Lagrangian method for nonconvex optimization with linear constraints. *J. Scientific Comp.*, 91(61), 2022.

[64] J. Zhang and Z.-Q. Luo. A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization. *SIAM Journal on Optimization*, 30(3):2272–2302, 2020.

[65] J. Zhang and Z.-Q. Luo. A global dual error bound and its application to the analysis of linearly constrained nonconvex optimization. *SIAM Journal on Optimization*, 32(3):2319–2346, 2022.