Efficient parameter-free restarted accelerated gradient methods for convex and strongly convex optimization

Arnesh Sujanani * Renato D.C. Monteiro †

October 5, 2024 (revised: October 11, 2024)

Abstract

This paper develops a new parameter-free restarted method, namely RPF-SFISTA, and a new parameter-free aggressive regularization method, namely A-REG, for solving strongly convex and convex composite optimization problems, respectively. RPF-SFISTA has the major advantage that it requires no knowledge of both the strong convexity parameter of the entire composite objective and the Lipschitz constant of the gradient. Unlike several other restarted first-order methods which restart an accelerated composite gradient (ACG) method after a predetermined number of ACG iterations have been performed, RPF-SFISTA checks a key inequality at each of iterations to determine when to restart. Extensive computational experiments show that RPF-SFISTA is roughly 3 to 15 times faster than other state-of-the-art restarted methods on four important classes of problems. The A-REG method, developed for convex composite optimization, solves each of its strongly convex regularized subproblems according to a stationarity criterion by using the RPF-SFISTA method with a possibly aggressive choice of initial strong convexity estimate. This scheme is thus more aggressive than several other regularization methods which solve their subproblems by running a standard ACG method for a predetermined number of iterations.

1 Introduction

This paper presents present new restarted and aggressive parameter-free first-order methods for solving the composite optimization problem

$$\phi_* := \min\{\phi(z) := f(z) + h(z)\}, \tag{1}$$

where ϕ is a function that is assumed to be either convex or $\bar{\mu}$ strongly convex, $h : \Re^n \to (-\infty, \infty]$ is a closed proper convex function, and $f : \Re^n \to \Re$ is a real-valued differentiable convex function whose gradient is $\bar{L}$–Lipschitz continuous.

The main focus of this paper is to propose a computationally efficient restarted parameter-free method, namely RPF-SFISTA, for solving strongly convex composite optimization (SCCO) problems. At each iteration, RPF-SFISTA proceeds as follows: it first calls a strongly convex accelerated composite gradient method (S-ACG) developed in [38] with an (usually aggressive)

*Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, N2L 3G1. (Email: a3sujana@uwaterloo.ca). This work was primarily done while this author was a PhD student at Georgia Institute of Technology. While at Georgia Institute of Technology, he was partially supported by AFORS Grant FA9550-22-1-0088.

†Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (Email: monteiro@isye.gatech.edu). This author was partially supported by AFORS Grant FA9550-22-1-0088.

estimate μ of the strong convexity parameter $\bar{\mu}$ until either a desired stationary point is obtained or a certain key inequality is violated; in the latter case S-ACG is invoked again with strong convexity estimate set to $\mu/2$ and initial point set to the best point obtained in the previous S-ACG call. Using this scheme, RPF-SFISTA restarts a logarithmic number of times and it achieves an iteration complexity of $\tilde{\mathcal{O}}\left(\sqrt{\bar{L}/\bar{\mu}}\right)$. RPF-SFISTA's superior numerical performance is also displayed through extensive computational experiments.

This paper also proposes a dynamic aggressive regularization method, namely A-REG, for solving convex composite optimization (CCO) problems. A-REG solves a sequence of strongly convex regularized subproblems using RPF-SFISTA with an aggressive choice of strong convexity estimate. Under the assumption of bounded sublevel sets, A-REG achieves a complexity of $\tilde{\mathcal{O}}\left(\sqrt{\bar{L}/\epsilon}\right)$, which is optimal up to logarithmic terms.

Literature review. We divide our discussion here into methods that were designed to solve SCCO and CCO problems, respectively.

Methods for SCCO: Methods that restart accelerated composite gradient (ACG) methods for solving SCCO problems have been proposed as early as 2008 in [15] (see also [16] for the published version) while methods that restart strongly convex variants of accelerated composite gradient (S-ACG) methods have been proposed as early as 2013 in [32]. Table 1 (resp. Table 2) below summarizes the differences between the existing restarted ACG (resp. S-ACG) methods.

We now briefly comment on each of the columns in both tables. The "Universality" column describes which parameters each method is universal with respect to. Methods that consider the setting where f is $\bar{\mu}_f$-strongly convex (resp. $\bar{L}$-smooth) and require no knowledge of $\bar{\mu}_f$ (resp. $\bar{L}$) are said to be $\bar{\mu}_f$-universal (resp. $\bar{L}$-universal). Likewise, methods that consider the setting where the entire composite function ϕ is assumed to be $\bar{\mu}$-strongly convex and that require no knowledge of $\bar{\mu}$ are said to be $\bar{\mu}$-universal. If an entry is marked with $*$, then it means that the method is not universal with respect to that parameter. The "Composite Objective" column displays whether the method considers a general composite objective as in (1) or the special case where h is the indicator function of a closed convex set (i.e., $h = \delta_C$). The "Stationarity" column indicates whether the method terminates according to a checkable stationarity termination criterion or according to a condition that involves the optimal value of (1) (which is usually unknown). The "Convergence Proof" column presents whether a method has any convergence guarantees or not. Finally, the "Restart Condition" column indicates whether the ACG (or S-ACG) restarts when a checkable condition is satisfied at some iteration or after a predetermined number of iterations is performed.

It has been observed in practice that restarted S-ACG methods tend to perform much better than restarted ACG methods. Also, methods that restart based on a checkable condition tend to have better computational performance than methods that restart based on a predetermined number of iterations.

Name	Universality	Composite Objective	Stationarity	Convergence Proof	Restart Condition
[15, 16]	$(*, *)$	No ($h = \delta_C$)	No	Yes	Predetermined
Sync||FOM (2022) [35]	$(\bar{\mu}_f, \bar{L})$	No ($h = \delta_C$)	No	Yes	Checkable
[1, 2, 3, 4, 10, 36]	$(\bar{\mu}, *)$	Yes	Yes	Yes	Predetermined
Free-FISTA (2023) [5]	$(\bar{\mu}, \bar{L})$	Yes	Yes	Yes	Predetermined

Table 1: Comparison of restarted ACG methods for SCCO.

This paragraph solely focuses on describing strongly convex variants of FISTA or S-ACG meth-

ods that have been previously developed. Methods for SCCO problems that use S-ACG and that are parameter-dependent and require knowledge of the strong convexity parameter underlying the objective function have been proposed in the following works [7, 9, 11, 12, 27, 29]. Universal restarted methods, including the RPF-SFISTA method in this paper, have also been proposed (see also [17, 23, 25, 32]). Table 2 below highlights the differences between each of the restarted S-ACG methods.

Name	Universality	Composite Objective	Stationarity	Convergence Proof	Restart Condition
[25, 32]	$(\bar{\mu}_f, \bar{L})$	Yes	Yes	Yes	Checkable
GR-FISTA (2022) [23]	$(\bar{\mu}_f, *)$	Yes	No	No	Checkable
SCAR (2023) [17]	$(\bar{\mu}_f, \bar{L})$	No	Yes	Yes	Checkable
RPF-SFISTA	$(\bar{\mu}, \bar{L})$	Yes	Yes	Yes	Checkable

Table 2: Comparison of RPF-SFISTA with other restarted S-ACG methods for SCCO.

As seen from Table 2, RPF-SFISTA is the only parameter-free $\bar{\mu}$-universal method that restarts a S-ACG method based on a checkable condition. It is observed in practice that methods that restart a S-ACG method based on a checkable condition tend to perform better in practice. We display through extensive computational experiments that RPF-SFISTA is roughly three to fifteen times faster than other state-of-the-art restarted methods on four important classes of problems. It is also worth mentioning that $\bar{\mu}$ can be much larger than $\bar{\mu}_f + \bar{\mu}_h$. Hence, methods that are $\bar{\mu}$-universal have the advantage that their complexities tend to be better than those of $\bar{\mu}_f$ and $\bar{\mu}_h$-universal methods and nonuniversal methods which depend on $\bar{\mu}_f + \bar{\mu}_h$.

Methods for CCO: Many direct methods based on proximal gradient, ACG, or FISTA methods have been developed to solve CCO problems (see [6, 18, 19, 21, 22, 26, 31, 32, 37]). Heuristic restart schemes that restart FISTA and that achieve good practical performance were also further proposed by O'Donoghue and Candès in [33]. Convergence proofs of their generic heuristic schemes have been provided only in very special cases [28].

In another line of research, regularization methods for solving CCO problems have been proposed as early as 2012 by Nesterov in [30]. In the unconstrained setting, Nesterov proposed a static regularization method where he applied a S-ACG method a single time to the regularized objective $\phi(z)+\delta\|z-z_0\|^2$. His method achieves a complexity of $\mathcal{O}\left(\sqrt{\bar{L}/\epsilon}\right)$ for finding an ϵ-stationary point, a complexity which has not yet been established by a direct method. Similar regularization schemes that are dynamic and that consider the more general composite setting have been proposed in [8, 13, 24, 34]. Roughly speaking, dynamic regularization methods solve a sequence of regularized subproblems of the form $\phi(z) + \delta_k\|z - z_k\|^2$. If the solution z_{k+1} of the k-th subproblem is not optimal for (1), the regularization factor δ_k is halved, the next prox center is updated to be z_{k+1}, and S-ACG is invoked again to solve the new subproblem. More recently, dynamic parameter-free regularization methods, including the A-REG method proposed in this paper and the methods in [14, 17], have also been proposed for solving CCO problems. Table 3 compares the features of A-REG with other regularization methods for CCO.

We briefly describe each of its columns. The first three columns describe the same features as the first three columns of Table 1, whose formal descriptions were given previously. The fourth column presents whether a method solves its strongly convex subproblems according to a stationarity condition that the ACG variant, which the method uses, checks at each of its iterations or whether the method solves each subproblem by running a predetermined number of ACG iterations. The fifth column displays whether a method uses a restarted S-ACG method with an usually aggressive

strong convexity estimate or a standard S-ACG method to solve its regularized subproblems. The last column presents whether a method is a static regularization method or a dynamic one that adaptively updates its prox-center and regularization parameter.

Name	Universality of Method	Composite Objective	Stationarity Termination	Checkable Condition for Subproblem	Restart S-ACG	Dynamic Regularization
Nesterov (2012) [30]	*	No	Yes	Yes	No	No
Catalyst (2015) [24]	*	Yes	No	No	No	Yes
[8, 13]	*	Yes	Yes	Yes	No	Yes
4WD-Catalyst (2018) [34]	*	Yes	Yes	No	No	Yes
APD Method (2022) [14]	$\bar{L}$	Yes	Yes	Yes	No	Yes
AR Method (2023) [17] [1]	$\bar{L}$	Yes	Yes	No	No	Yes
A-REG	$\bar{L}$	Yes	Yes	Yes	Yes	Yes

Table 3: Comparison of A-REG with other regularization methods for CCO.

As can be seen from Table 3, A-REG is the only method that solves its dynamic regularization subproblems using a restart S-ACG method which makes an adaptive choice of strong convexity parameter.

Organization of the paper. Subsection 1.1 presents basic definitions and notations used throughout this paper. Section 2 formally describes the RPF-SFISTA method for solving SCCO problems and its main complexity result and analysis. Section 3 presents the A-REG method for solving CCO problems and its main complexity result and analysis. Section 4 presents extensive computational experiments that display the superior numerical performance of RPF-SFISTA compared to other state-of-the-art restart schemes on four different important classes of composite optimization problems. Finally, Appendix A is dedicated to proving an important proposition used in the complexity analysis of RPF-SFISTA.

1.1 Basic Definitions and Notations

This subsection presents notation and basic definitions used in this paper.

Let $\Re_+$ and $\Re_{++}$ denote the set of nonnegative and positive real numbers, respectively. We denote by $\Re^n$ an n-dimensional inner product space with inner product and associated norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. We use $\Re^{l \times n}$ to denote the set of all $l \times n$ matrices and $\mathbb{S}^+_n$ to denote the set of positive semidefinite matrices in $\Re^{n \times n}$. The smallest positive singular value of a nonzero linear operator $Q : \Re^n \to \Re^l$ is denoted by ν^+_Q. For a given closed convex set $Z \subset \Re^n$, its boundary is denoted by ∂Z and the distance of a point $z \in \Re^n$ to Z is denoted by $\mathrm{dist}(z, Z)$. 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 any $t > 0$ and $b \geq 0$, we let $\log^+_b(t) := \max\{\log t, b\}$, and we define $\mathcal{O}_1(\cdot) = \mathcal{O}(1 + \cdot)$.

The domain of a function $h : \Re^n \to (-\infty, \infty]$ is the set $\mathrm{dom}\, h := \{x \in \Re^n : h(x) < +\infty\}$. Moreover, h is said to be proper if $\mathrm{dom}\, h \neq \emptyset$. The set of all lower semi-continuous proper convex functions defined in $\Re^n$ is denoted by $\overline{\mathrm{Conv}}\, \Re^n$. The ε-subdifferential of a proper function $h : \Re^n \to (-\infty, \infty]$ is defined by

$$\partial_\varepsilon h(z) := \{u \in \Re^n : h(z') \geq h(z) + \langle u, z' - z \rangle - \varepsilon, \quad \forall z' \in \Re^n\} \tag{2}$$

[1] In the general composite setting, the AR method in [17] is not developed or presented as a parameter-free method and requires knowledge of the Lipschitz constant, $\bar{L}$. However, in the unconstrained setting the authors develop a parameter-free variant of the AR method, which they say can be extended to the general composite setting.

for every $z \in \Re^n$. The classical subdifferential, denoted by $\partial h(\cdot)$, corresponds to $\partial_0 h(\cdot)$. Recall that, for a given $\varepsilon \geq 0$, the ε-normal cone of a closed convex set C at $z \in C$, denoted by $N_C^\varepsilon(z)$, is

$$N_C^\varepsilon(z) := \{\xi \in \Re^n : \langle \xi, u - z \rangle \leq \varepsilon, \quad \forall u \in C\}.$$

The normal cone of a closed convex set C at $z \in C$ is denoted by $N_C(z) = N_C^0(z)$. If ϕ is a real-valued function which is differentiable at $\bar{z} \in \Re^n$, then its affine approximation $\ell_\phi(\cdot, \bar{z})$ at $\bar{z}$ is given by

$$\ell_\phi(z; \bar{z}) := \phi(\bar{z}) + \langle \nabla \phi(\bar{z}), z - \bar{z} \rangle \quad \forall z \in \Re^n. \tag{3}$$

2 Strongly Convex Composite Optimization (SCCO)

This section presents a restarted parameter-free FISTA variant, namely RPF-SFISTA, for solving strongly convex composite optimization (SCCO) problems.

Specifically, RPF-SFISTA assumes that problem (1) has an optimal solution z^* and that functions f, h, and ϕ satisfy the following assumptions:

(A1) $f : \mathbb{E} \to \Re$ is a differentiable convex function that is $\bar{L}$-smooth, i.e., there exists $\bar{L} \geq 0$ such that, for all $z, z' \in \mathbb{E}$,

$$\|\nabla f(z') - \nabla f(z)\| \leq \bar{L}\|z' - z\|; \tag{4}$$

(A2) $h : \mathbb{E} \to \Re \cup \{+\infty\}$ is a possibly nonsmooth convex function with domain denoted by $\mathcal{H}$;

(A3) ϕ is a $\bar{\mu}$-strongly convex function, where $\bar{\mu} > 0$.

We now describe the type of approximate solution that RPF-SFISTA aim to find.

Given ϕ satisfying the above assumptions and a tolerance parameter $\hat{\epsilon} > 0$, the goal of RPF-SFISTA is to find a pair $(y, v) \in \mathcal{H} \times \mathbb{E}$ such that

$$\|v\| \leq \hat{\epsilon}, \quad v \in \nabla f(y) + \partial h(y). \tag{5}$$

Any pair (y, v) satisfying (5) is said to be an **ϵ-optimal solution** of (1).

The rest of this section is broken up into three subsections. The first one motivates and states the RPF-SFISTA method and presents its main complexity results. The second and third subsections present the proofs of the main results.

2.1 The RPF-SFISTA method

This subsection motivates and states the RPF-SFISTA method and presents its main complexity results.

The RPF-SFISTA is essentially a restarted version of a S-ACG variant (see for example [27]) that also performs backtracking line-search for the smoothness parameter. RPF-SFISTA calls the S-ACG variant with an aggressive estimate μ for the strong convexity parameter $\bar{\mu}$. A novel condition is then checked at each of the variant's iterations to see if RPF-SFISTA should restart and call the variant again with a smaller estimate $\mu = \mu/2$. Each time RPF-SFISTA restarts, a new cycle of RPF-SFISTA is said to begin. If RPF-SFISTA calls the S-ACG variant with a strong convexity estimate μ such that $\mu \in (0, \bar{\mu}]$, then it is shown in Proposition 2.1 below that the current cycle of RPF-SFISTA must terminate with a pair (y, v) that satisfies (5) and is thus an ϵ-approximate solution of (1). Hence, RPF-SFISTA performs at most a logarithmic number of restarts/cycles. The formal description of RPF-SFISTA algorithm is now presented.

RPF-SFISTA Method

Universal Parameters: scalars $\chi \in (0,1)$ and $\beta > 1$.
Inputs: let scalars $\left(\mu_0, \bar{M}_0\right) \in \Re^2_{++}$, an initial point $z_0 \in \mathcal{H}$, a tolerance $\hat{\epsilon} > 0$, and functions (f, h) and $\phi := f + h$ be given, and set $l = 1$.
Output: a quadruple (y, v, ξ, L).

0. set $j = 1$, initial point $x_0 = z_{l-1}$, estimates $\underline{\mathrm{M}}_l \in [\max\{0.25\bar{M}_{l-1}, \bar{M}_0\}, \bar{M}_{l-1}]$ and $\mu = \mu_{l-1}$, points $(\xi_0, y_0) = (x_0, x_0)$, and scalars $(A_0, \tau_0, L_0) = (0, 1, \underline{\mathrm{M}}_l)$;

1. set $L_j = L_{j-1}$;

2. compute

$$a_{j-1} = \frac{\tau_{j-1} + \sqrt{\tau_{j-1}^2 + 4\tau_{j-1}A_{j-1}L_j}}{2L_j}, \quad \tilde{x}_{j-1} = \frac{A_{j-1}y_{j-1} + a_{j-1}x_{j-1}}{A_{j-1} + a_{j-1}}, \tag{6}$$

$$y_j := \operatorname*{argmin}_{u \in \mathcal{H}} \left\{ q_{j-1}(u; \tilde{x}_{j-1}, L_j) := \ell_f(u; \tilde{x}_{j-1}) + h(u) + \frac{L_j}{2}\|u - \tilde{x}_{j-1}\|^2 \right\}; \tag{7}$$

if the inequality

$$\ell_f(y_j; \tilde{x}_{j-1}) + \frac{(1-\chi)L_j}{4}\|y_j - \tilde{x}_{j-1}\|^2 \geq f(y_j) \tag{8}$$

holds go to step 3; else set $L_j \leftarrow \beta L_j$ and repeat step 2;

3. compute

$$\xi_j = \begin{cases} y_j & \text{if } \phi(y_j) \leq \phi(\xi_{j-1}) \\ \xi_{j-1} & \text{otherwise,} \end{cases} \tag{9}$$

$$A_j = A_{j-1} + a_{j-1}, \quad \tau_j = \tau_{j-1} + \frac{a_{j-1}\mu}{2}, \tag{10}$$

$$s_j = L_j(\tilde{x}_{j-1} - y_j), \tag{11}$$

$$x_j = \frac{1}{\tau_j}\left[\frac{\mu a_{j-1} y_j}{2} + \tau_{j-1}x_{j-1} - a_{j-1}s_j\right], \tag{12}$$

$$v_j = \nabla f(y_j) - \nabla f(\tilde{x}_{j-1}) + s_j; \tag{13}$$

4. if the inequality

$$\|\xi_j - x_0\|^2 \geq \chi A_j L_j \|y_j - \tilde{x}_{j-1}\|^2, \tag{14}$$

holds, then go to step 5; otherwise **restart**, set $z_l = \xi_j$, $\bar{M}_l = L_j$, $\mu_l = \mu/2$, and $l \leftarrow l + 1$, and go to step 0;

5. if the inequality

$$\|v_j\| \leq \hat{\epsilon} \tag{15}$$

holds then **stop** and output quadruple $(y, v, \xi, L) := (y_j, v_j, \xi_j, L_j)$; otherwise, set $j \leftarrow j + 1$ and go to step 1.

Several remarks about RPF-SFISTA are now given. First, RPF-SFISTA is an adaptive and parameter-free method in that it requires no knowledge of the Lipschitz and strong convexity parameters and instead adaptively performs line searches for these constants. Second, it performs two types of iterations, namely cycles indexed by l and inner ACG/FISTA iterations which are indexed by j. The number of restarts RPF-SFISTA performs is equivalent to the number of cycles it performs minus one. Third, steps 2 and 3 of RPF-SFISTA are essentially equivalent to an iteration of a S-ACG variant that performs a line-search for the Lipschitz constant of the gradient. Fourth, step 4 of RPF-SFISTA checks a novel condition (14) to determine whether it should restart or not. If relation (14) fails to hold during an iteration of the l-th cycle of RPF-SFISTA, RPF-SFISTA restarts and begins the $(l+1)$st cycle with a smaller estimate for the strong convexity parameter μ. Fifth, RPF-SFISTA performs warm-restarting, i.e., when RPF-SFISTA restarts and begins the $(l+1)$st cycle it takes as initial point for this cycle the point with the best function value that was found during the previous cycle. Finally, it will be shown that for any $j \geq 1$, the pair (y_j, v_j) always satisfies the inclusion in (5). As a consequence, if RPF-SFISTA stops in step 5, output pair (y, v) is an ϵ-optimal solution of (1).

Before stating the main results of the RPF-SFISTA method, the following quantities are introduced

$$C_{\bar{\mu}}(\cdot) := \frac{8}{\bar{\mu}} \left[\phi(\cdot) - \phi(z^*)\right], \quad \kappa := 2\beta/(1-\chi), \tag{16}$$

$$\zeta_l := \bar{L} + \max\{\underline{M}_l, \kappa\bar{L}\}, \quad Q_l := 2\sqrt{2}\sqrt{\frac{\max\{\underline{M}_l, \kappa\bar{L}\}}{\mu_{l-1}}}, \tag{17}$$

where $\bar{L}$ is as in (4) and z^* is an optimal solution of (1).

The following proposition and theorem state the main complexity results of the RPF-SFISTA method and key properties of its output. The proofs of Proposition 2.1 and Theorem 2.2 are given in Appendix A and Subsection 2.2, respectively.

Proposition 2.1. *The following statements about the l-th cycle of RPF-SFISTA hold:*

(a) it stops (in either step 4 or step 5) in at most

$$\left\lceil (1+Q_l)\log_1^+\left(\frac{C_{\bar{\mu}}(z_{l-1})\zeta_l^2}{\chi\hat{\epsilon}^2}\right) + 1\right\rceil + \left\lceil \frac{\log_0^+(2\bar{L}/((1-\chi)\underline{M}_l))}{\log\beta}\right\rceil \tag{18}$$

ACG iterations/resolvent evaluations where χ and β are input parameters to RPF-SFISTA, $\hat{\epsilon}$ is the input tolerance, $\bar{L}$ is as in (4), $C_{\bar{\mu}}(\cdot)$ and κ are as in (16), and Q_l and ζ_l are as in (17);

(b) if the cycle terminates in its step 5, then it outputs a quadruple (y, v, ξ, L) that satisfies

$$\phi(\xi) \leq \min\left\{\phi(z_0), \phi(y)\right\}, \quad \bar{M}_0 \leq L \leq \max\left\{\bar{M}_0, \kappa\bar{L}\right\} \tag{19}$$

and such that (y, v) is an $\hat{\epsilon}$-optimal solution of (1), where z_0 is the initial point and $\bar{M}_0$ is an input to RPF-SFISTA;

(c) if $\mu_{l-1} \in (0, \bar{\mu}]$, then the cycle always stops successfully in step 5 with a quadruple (y, v, ξ, L) that satisfies (19) and such that (y, v) is an $\hat{\epsilon}$-optimal solution of (1) in at most (18) ACG iterations/resolvent evaluations.

The following theorem states the main complexity result of RPF-SFISTA and key properties of its output.

Theorem 2.2. *RPF-SFISTA terminates with a quadruple (y, v, ξ, L) that satisfies (19) and such that (y, v) is an $\hat{\epsilon}$-optimal solution of (1) in at most*

$$\mathcal{O}_1\left(\left\lceil \log_1^+(2\mu_0/\bar{\mu})\right\rceil \left[\sqrt{\frac{\max\{\bar{M}_0, \bar{L}\}}{\min\{\mu_0, \bar{\mu}/2\}}}\log_1^+\left(\frac{(\bar{L}^2+\bar{M}_0^2)C_{\bar{\mu}}(z_0)}{\hat{\epsilon}^2}\right)+\log_0^+(\bar{L}/\bar{M}_0)\right]\right) \tag{20}$$

ACG iterations/resolvent evaluations where $\hat{\epsilon}$, μ_0, and $\bar{M}_0$ are inputs to RPF-SFISTA, z_0 is the initial point, and $\bar{\mu}$, $C_{\bar{\mu}}(\cdot)$, and $\bar{L}$ are as in (B2), (16), and (4), respectively.

A remark about Theorem 2.2 is now given. If $\mu_0 = \Omega(\bar{\mu})$ and $\bar{M}_0 = \Omega(\bar{L})$, it then follows from the above result and the definition of $C_{\bar{\mu}}(\cdot)$ in (16) that RPF-SFISTA performs at most

$$\mathcal{O}_1\left(\sqrt{\frac{\bar{L}}{\bar{\mu}}}\log_1^+\left(\frac{\bar{L}^2}{\bar{\mu}\hat{\epsilon}^2}\right)\right)$$

ACG iterations/resolvent iterations to find a pair (y, v) that is an ϵ-approximate optimal solution of (1).

2.2 Proof of Theorem 2.2

This subsection is dedicated to proving Theorem 2.2. The following two lemmas present key properties of the iterates generated during the l-th cycle of RPF-SFISTA. The proof of the first lemma below is not given as it closely resembles the proofs of Lemmas A.3 and A.4 in [38].

Lemma 2.3. *Let κ be as in (16) and ζ_l and Q_l be as in (17). For every iteration index $j \geq 1$ generated during the l-th cycle of RPF-SFISTA, the following statements hold:*

(a) $\{L_j\}$ is nondecreasing;

(b) the following relations hold

$$\tau_{j-1} = 1 + \frac{\mu A_{j-1}}{2}, \quad \frac{\tau_{j-1}A_j}{a_{j-1}^2} = L_j, \tag{21}$$

$$\underline{M}_l \leq L_{j-1} \leq \max\{\underline{M}_l, \kappa\bar{L}\}, \tag{22}$$

$$v_j \in \nabla f(y_j) + \partial h(y_j), \quad \|v_j\| \leq \zeta_l\|y_j - \tilde{x}_{j-1}\|; \tag{23}$$

(c) it holds that

$$A_j L_j \geq \max\left\{\frac{j^2}{4}, \left(1+Q_l^{-1}\right)^{2(j-1)}\right\}. \tag{24}$$

Lemma 2.4. *For every iteration index $j \geq 1$ generated during the l-th cycle of RPF-SFISTA, it holds that $\phi(\xi_j) \leq \min\{\phi(\xi_{j-1}), \phi(y_j)\}$. As a consequence, the following relation holds*

$$\phi(\xi_j) \leq \phi(z_{l-1}). \tag{25}$$

Proof. Let $j \geq 1$ be an iteration index generated during the l-th cycle of RPF-SFISTA. To see that $\phi(\xi_j) \leq \min\{\phi(\xi_{j-1}), \phi(y_j)\}$, consider two possible cases. First, suppose that $\phi(y_j) \leq \phi(\xi_{j-1})$. It follows from the update rule for ξ_j in (9) that $\xi_j = y_j$ and hence that $\phi(\xi_j) = \phi(y_j) \leq \phi(\xi_{j-1})$. For

the other case, suppose that $\phi(y_j) > \phi(\xi_{j-1})$. Relation (9) then immediately implies that $\xi_j = \xi_{j-1}$ and hence that $\phi(\xi_j) = \phi(\xi_{j-1}) < \phi(y_j)$. Combining the two cases proves the inequality holds.

It then follows from this inequality, a simple induction argument, and the fact that $\xi_0 = x_0$ that $\phi(\xi_j) \leq \phi(x_0)$. This relation and the fact that x_0 is set as z_{l-1} at the beginning of the l-th cycle of RPF-SFISTA immediately imply relation (25). □

Lemma 2.5. *For any cycle index $l \geq 1$ generated by RPF-SFISTA, the quantity $\underline{M}_l$ set in step 0 satisfies*

$$\underline{M}_l \leq \max\left\{\bar{M}_0, \kappa\bar{L}\right\} \tag{26}$$

where $\bar{M}_0 > 0$ is an input to RPF-SFISTA and $\bar{L}$ and κ are as in (4) and (16), respectively.

Proof. First, we show that for any cycle index $l \geq 2$ generated by RPF-SFISTA, the following relation

$$\bar{M}_0 \leq \bar{M}_{l-1} \leq \max\left\{\underline{\mathrm{M}}_{l-1}, \kappa\bar{L}\right\} \tag{27}$$

holds. Since $l \geq 2$ is a cycle index generated by RPF-SFISTA, this implies that its $(l-1)$-st cycle terminates in its step 4 and hence $\bar{M}_{l-1}$ is generated. Both relations in (27) then immediately follow from this observation, the fact that the way $\bar{M}_{l-1}$ is chosen in step 0 implies that $\bar{M}_0 \leq \underline{\mathrm{M}}_{l-1}$, the fact that $\bar{M}_{l-1}$ is set in step 4 of RPF-SFISTA as L_j for some iteration index $j \geq 1$ generated during the $(l-1)$-st cycle, and the first and second inequalities in (22) with $l = l-1$.

The proof of (26) now follows from an induction argument. The result with $l = 1$ follows immediately from the fact that $\underline{\mathrm{M}}_1 = \bar{M}_0$. Suppose now that $l \geq 2$ is a cycle index generated by RPF-SFISTA and that inequality (26) holds for $l-1$. It follows from the way $\underline{\mathrm{M}}_l$ is chosen in step 0 at the beginning of the l-th cycle of RPF-SFISTA and the first relation in (27) that $\underline{\mathrm{M}}_l \leq \bar{M}_{l-1}$. This relation and the second relation in (27) then imply that

$$\underline{\mathrm{M}}_l \leq \bar{M}_{l-1} \leq \max\left\{\underline{\mathrm{M}}_{l-1}, \kappa\bar{L}\right\} \leq \max\left\{\bar{M}_0, \kappa\bar{L}\right\}$$

where the last inequality is due to the induction hypothesis. Hence, Lemma 2.5 holds. □

The following lemma establishes a bound on the quantity $C_{\bar{\mu}}(z_l)$ in terms of the initial point z_0.

Lemma 2.6. *For every cycle index $l \geq 1$ generated by RPF-SFISTA, the following relations hold*

$$\phi(z_{l-1}) \leq \phi(z_0) \tag{28}$$
$$C_{\bar{\mu}}(z_{l-1}) \leq C_{\bar{\mu}}(z_0) \tag{29}$$

where z_0 is the initial point of RPF-SFISTA and $C_{\bar{\mu}}(\cdot)$ is as in (16).

Proof. Both relations clearly hold for $l = 1$, so let $l \geq 2$ be a cycle index generated by RPF-SFISTA. Since $l \geq 2$ is a cycle index generated by RPF-SFISTA, this implies that its $(l-1)$-st cycle terminates in step 4 and hence z_{l-1} is generated. It follows from this observation, the fact that z_{l-1} is set at the end of step of RPF-SFISTA as ξ_j for some iteration index $j \geq 1$ generated during the $(l-1)$-st cycle, and relation (25) with $l = l-1$ that $\phi(z_{l-1}) \leq \phi(z_{l-2})$. This relation and a simple induction argument then immediately imply relation (28). Relation (29) then follows from relation (28) and the definition of $C_{\bar{\mu}}(\cdot)$ in (16). □

The following proposition establishes an upper bound on the number of cycles that RPF-SFISTA performs and a bound on the number of iterations each cycle performs.

Proposition 2.7. *The following statements about RPF-SFISTA hold:*

(a) RPF-SFISTA performs at most $\lceil \log_1^+(2\mu_0/\bar{\mu}) \rceil$ *cycles to find a quadruple* (y, v, ξ, L) *that satisfies relation (19) and such that* (y, v) *is an* $\hat{\epsilon}$*-optimal solution of (1). Moreover, for every cycle index* $l \geq 1$ *generated by RPF-SFISTA, it holds that* $\mu_{l-1} \geq \min\{\mu_0, \bar{\mu}/2\}$*;*

(b) each cycle of RPF-SFISTA performs at most

$$\mathcal{O}_1 \left(\sqrt{\frac{\max\{\bar{M}_0, \bar{L}\}}{\min\{\mu_0, \bar{\mu}/2\}}} \log_1^+ \left(\frac{(\bar{L}^2 + \bar{M}_0^2) C_{\bar{\mu}}(z_0)}{\hat{\epsilon}^2} \right) + \log_0^+(\bar{L}/\bar{M}_0) \right)$$

ACG iterations/resolvent evaluations where $\hat{\epsilon}$ *and* μ_0 *are inputs to RPF-SFISTA,* z_0 *is the initial point, and* $\bar{\mu}$*,* $\bar{L}$*, and* $C_{\bar{\mu}}(\cdot)$ *are as in (B2), (4), and (16), respectively.*

Proof. (a) It follows immediately from Proposition 2.1(c) that the l-th cycle of RPF-SFISTA always terminates successfully in step 5 with a quadruple (y, v, ξ, L) that satisfies relation (19) and such that (y, v) is an $\hat{\epsilon}$-optimal solution of (1), if it is performed with $\mu_{l-1} \in (0, \bar{\mu}]$. Both conclusions of (a) then follow immediately from this observation, from the way that μ_l is updated when a cycle terminates in step 4 of RPF-SFISTA, and the fact that the first cycle of RPF-SFISTA is performed with $\mu = \mu_0$.

(b) Lemma 2.5 implies that $\underline{M}_l \leq \max\{\bar{M}_0, \kappa\bar{L}\}$ for every cycle index $l \geq 1$ generated by RPF-SFISTA. This relation and the definition of ζ_l in (17) imply that $\zeta_l^2 \leq \max\{(\bar{L} + M_0)^2, (\kappa + 1)^2\bar{L}^2\}$ and hence that $\zeta_l^2 = \mathcal{O}(\bar{L}^2 + \bar{M}_0^2)$. These relations, the fact that the way $\underline{M}_l$ is chosen in step 0 implies that $\underline{M}_l \geq \bar{M}_0$, Proposition 2.1(a), and the definition of Q_l imply that the l-th cycle performs at most

$$\mathcal{O}_1 \left(\sqrt{\frac{\max\{\bar{M}_0, \bar{L}\}}{\mu_{l-1}}} \log_1^+ \left(\frac{(\bar{L}^2 + \bar{M}_0^2) C_{\bar{\mu}}(z_{l-1})}{\hat{\epsilon}^2} \right) + \log_0^+(\bar{L}/\bar{M}_0) \right)$$

ACG iterations/resolvent evaluations. The result then follows from the above relation, relation (29), and the last conclusion of Proposition 2.7(a). □

We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2. The first conclusion of Proposition 2.7(a) and Proposition 2.7(b) immediately imply the result. □

3 Convex Composite Optimization (CCO)

This section presents an aggressive regularized method, namely A-REG, for solving convex composite optimization (CCO) problems. Specifically, A-REG considers the problem

$$\min\{\psi(u) := \psi_s(u) + \psi_n(u) : u \in \mathbb{E}\} \tag{30}$$

whose solution set is nonempty and where functions ψ, ψ_s, and ψ_n are assumed to satisfy the following assumptions:

(B1) $\psi_n : \mathbb{E} \to \Re \cup \{+\infty\}$ is a possibly nonsmooth convex function with domain denoted by $\mathcal{N}$;

(B2) $\psi_s : \mathbb{E} \to \Re$ is a differentiable convex function that is $\bar{L}_{\psi_s}$-smooth, i.e., there exists $\bar{L}_{\psi_s} \geq 0$ such that, for all $z, z' \in \mathbb{E}$,

$$\|\nabla\psi_s(z') - \nabla\psi_s(z)\| \leq \bar{L}_{\psi_s}\|z' - z\|; \tag{31}$$

(B3) the sublevel sets of ψ are bounded.

Assumption (B3) is a common assumption and a similar type of assumption is made in [17, 24, 30].

Given a tolerance $\epsilon > 0$, the goal of A-REG is to find a pair $(w, r) \in \mathcal{N} \times \mathbb{E}$ such that

$$\|r\| \leq \epsilon, \quad r \in \nabla\psi_s(w) + \partial\psi_n(w). \tag{32}$$

Any pair (w, r) satisfying (32) is said to be an **ϵ-optimal solution** of (30).

3.1 The Aggressive Regularization (A-REG) method

This subsection motivates and states the aggressive regularization (A-REG) method and presents its main complexity results.

Like the methods in [14, 17, 24, 30, 34], A-REG is a regularization method. At each of iterations, A-REG forms strongly convex regularized subproblems which it solves using the RPF-SFISTA method developed in Subsection 2.1. A-REG calls RPF-SFISTA with an aggressive initial choice of strong convexity estimate, μ_0, that is possibly much larger than the known strong convexity parameter of the objective of the regularized subproblem. RPF-SFISTA checks a key inequality at each of iterations to determine when a regularized subproblem has been approximately solved. Hence, A-REG is more aggressive than the schemes employed by [17, 24, 34], which run a standard ACG or S-ACG variant for a predetermined number of iterations to solve each subproblem. The schemes in [14, 30] do not run a S-ACG variant for a predetermined number of iterations to solve each subproblem, but are less aggressive than A-REG as both schemes do not use a restarted S-ACG method for solving each of subproblem but rather they use a standard S-ACG method with an accurate estimate for strong convexity parameter to do so.

The A-REG method is now presented.

A-REG Method

Universal Parameters: scalars $\chi \in (0, 1)$ and $\beta > 1$.
Inputs: let scalars $B \geq 1$ and $(\delta_0, \bar{N}_0) \in \Re^2_{++}$, an initial point $\vartheta_0 \in \mathcal{N}$, a tolerance $\epsilon > 0$, and functions (ψ_s, ψ_n) and $\psi := \psi_s + \psi_n$ be given, and set $w_0 = \vartheta_0$ and $k = 1$.
Output: a pair (w, r) satisfying (32).

1. choose $\underline{N}_k \in [\max\{0.25\bar{N}_{k-1}, \bar{N}_0\}, \bar{N}_{k-1}]$ and call the RPF-SFISTA method described in Subsection 2.1 with inputs

$$z_0 = \vartheta_{k-1}, \quad (\mu_0, \bar{M}_0) = (B\delta_{k-1}, \underline{N}_k), \quad \hat{\epsilon} = \epsilon/6 \tag{33}$$

$$(f, h) = \left(\psi_s + \frac{\delta_{k-1}}{2}\|\cdot - \vartheta_{k-1}\|^2, \psi_n\right), \quad \phi(\cdot) = \psi(\cdot) + \frac{\delta_{k-1}}{2}\|\cdot - \vartheta_{k-1}\|^2 \tag{34}$$

and let $(w_k, u_k, \vartheta_k, \bar{N}_k)$ denote its output (y, v, ξ, L);

2. set

$$r_k := u_k + \delta_{k-1}(\vartheta_{k-1} - w_k); \tag{35}$$

3. if $\|r_k\| \leq \epsilon$ then stop and output $(w, r) = (w_k, r_k)$; else set $\delta_k = \delta_{k-1}/2$, $k \leftarrow k + 1$, and go to step 1.

Several remarks about A-REG are now given. First, A-REG is a parameter-free and an aggressive method in that it requires no knowledge of the Lipschitz constant and employs a restarted parameter-free method, RPF-SFISTA, to solve its strongly convex subproblems. Second, the k-th iteration of A-REG calls RPF-SFISTA with a function ϕ as in (34) that is δ_{k-1}-strongly convex. However, A-REG calls RPF-SFISTA with an aggressive initial strong convexity estimate $\mu_0 = B\delta_{k-1}$ where $B \geq 1$. Hence, A-REG differs from the methods in [14, 17, 24, 30, 34] as these methods **do not call restarted** S-ACG methods with an aggressive initial choice for the strong convexity estimate to solve their subproblems, but rather they call standard S-ACG or even ACG methods to solve them. Finally, it will be shown in the next subsection that for any $k \geq 1$, the pair (w_k, r_k) always satisfies the inclusion $r_k \in \nabla\psi_s(w_k) + \partial\psi_n(w_k)$. Thus, if A-REG stops in its step 3, it follows that output pair $(w, r) = (w_k, r_k)$ is an ϵ-optimal solution of (30).

Remark 3.1. *Assumption (B3) implies that the sublevel set $S := \{x \in \mathcal{N} : \psi(x) \leq \psi(\vartheta_0)\}$ is bounded, i.e., any $\vartheta \in S$ satisfies $\|\vartheta\| \leq D$ where $D > 0$.*

Before stating the main result of the A-REG method, the following quantities are introduced

$$\bar{L}_{\delta_0} = \kappa\left(\bar{L}_{\psi_s} + \delta_0\right), \quad \psi^0 = \psi(\vartheta_0) - \psi(w^*), \quad \bar{\mathcal{L}}^2 = \max\left\{\bar{L}_{\delta_0}^2 + \bar{N}_0^2, 2\bar{L}_{\delta_0}^2\right\} \tag{36}$$

where κ is as in (16), δ_0 is an input parameter of A-REG, ϑ_0 is the initial point, and w^* is an optimal solution of (30).

The following theorem states the main complexity result of A-REG.

Theorem 3.1. *A-REG terminates with a pair (w, r) that is an ϵ-optimal solution of (30) in at most*

$$\mathcal{O}_1\left(\left\lceil \log_1^+\left(8D\delta_0/\epsilon\right)\right\rceil \left\lceil \log_1^+ 2B\right\rceil \left[\sqrt{\frac{2\max\left\{\bar{N}_0, \bar{L}_{\delta_0}\right\}}{\min\left\{\delta_0, \epsilon/(8D)\right\}}} \log_1^+\left(\frac{288\psi^0\bar{\mathcal{L}}^2}{\min\left\{\delta_0, \epsilon/(8D)\right\}\epsilon^2}\right) + \log(\bar{L}_{\delta_0}/\bar{N}_0)\right]\right)$$

ACG iterations/resolvent evaluations where ϵ, δ_0, and $\bar{N}_0$ are inputs to A-REG, $\bar{L}_{\delta_0}$, ψ^0, and $\bar{\mathcal{L}}^2$ are as in (36), and $B \geq 1$ and $D > 0$ are scalars as in (33) and Assumption 3.1, respectively.

Several remarks about Theorem 3.1 are now given. It follows from the definition of $\bar{L}_{\delta_0}$ in (36) that, up to logarithmic terms, A-REG performs at most

$$\mathcal{O}_1\left(\sqrt{\frac{\bar{L}_{\psi_s}}{\epsilon}}\right)$$

ACG iterations/resolvent iterations to find a pair (w, r) that is an ϵ-approximate optimal solution of (30). Ignoring logarithmic terms, this complexity is optimal for finding an approximate optimal solution according to the criterion in (32).

3.2 Proof of Theorem 3.1

This subsection is dedicated to proving Theorem 3.1. Before stating the next lemma, we introduce the following quantities which are used throughout this subsection

$$\psi_\delta(w; \vartheta) := \psi(w) + \frac{\delta}{2}\|w - \vartheta\|^2, \quad \mathcal{C}_\delta(\cdot) := \frac{8}{\delta}\left[\psi(\cdot) - \psi(w^*)\right] \tag{37}$$

where $\delta > 0$ is a scalar and w^* is an optimal solution of (30).

Since A-REG calls the RPF-SFISTA method during each of its iterations, the following lemma specializes Theorem 2.2, which states the complexity of RPF-SFISTA and key properties of its output, to this set-up.

Lemma 3.2. *The following statements about the k-th iteration of A-REG hold*

(a) the function f in (34) is $(\bar{L}_{\psi_s} + \delta_0)$-smooth and δ_{k-1}-strongly convex. As a consequence, the function, ϕ in (34) is δ_{k-1}-strongly convex;

(b) the call made to the RPF-SFISTA method in step 1 outputs a quadruple $(w_k, u_k, \vartheta_k, \bar{N}_k)$ that satisfies the following relations

$$\psi(\vartheta_k) \leq \psi(\vartheta_{k-1}), \quad \psi_{\delta_{k-1}}(\vartheta_k; \vartheta_{k-1}) \leq \psi_{\delta_{k-1}}(w_k; \vartheta_{k-1}), \quad \bar{N}_0 \leq \bar{N}_k \leq \max\{\underline{N}_k, \bar{L}_{\delta_0}\}, \tag{38}$$

$$u_k \in \nabla\psi_s(w_k) + \delta_{k-1}(w_k - \vartheta_{k-1}) + \partial\psi_n(w_k), \quad \|u_k\| \leq \frac{\epsilon}{6} \tag{39}$$

where $\bar{L}_{\delta_0}$ and $\psi_\delta(w; \vartheta)$ are as in (36) and (37), respectively;

(c) the call made to the RPF-SFISTA method performs at most

$$\mathcal{O}_1\left(\left\lceil \log_1^+(2B)\right\rceil \left[\sqrt{\frac{2\max\left\{\underline{N}_k, \bar{L}_{\delta_0}\right\}}{\delta_{k-1}}} \log_1^+\left(\frac{36\left(\bar{L}_{\delta_0}^2 + \underline{N}_k^2\right)\mathcal{C}_{\delta_{k-1}}(\vartheta_{k-1})}{\epsilon^2}\right) + \log(\bar{L}_{\delta_0}/\bar{N}_0)\right]\right)$$

ACG iterations/resolvent evaluations to find a quadruple $(w_k, u_k, \vartheta_k, \bar{N}_k)$ satisfying relations (38) and (39) , where $B \geq 1$ and $\mathcal{C}_\delta(\cdot)$ are as in (33) and (37), respectively.

Proof. (a) It follows immediately from the facts that ψ_s is $\bar{L}_{\psi_s}$-smooth and ψ_s and ψ are convex functions and the fact that during the k-th iteration of A-REG, the call to RPF-SFISTA is made with $f(\cdot) := \psi_s(\cdot) + 0.5\delta_{k-1}\|\cdot - \vartheta_{k-1}\|^2$ and $\phi(\cdot) := \psi(\cdot) + 0.5\delta_{k-1}\|\cdot - \vartheta_{k-1}\|^2$ that f is $(\bar{L}_{\psi_s} + \delta_{k-1})$-smooth and δ_{k-1}-strongly convex and ϕ is δ_{k-1}-strongly convex. The result then follows immediately from this conclusion and the fact that the way δ_k is updated in step 3 of A-REG implies that $\delta_k \leq \delta_0$ for any iteration index $k \geq 1$.

(b) It follows from the definitions of ϕ and ψ_δ in (34) and (37), respectively, that the call to RPF-SFISTA during the k-th iteration of A-REG is made with $\phi(\cdot) := \psi_{\delta_{k-1}}(\cdot; \vartheta_{k-1})$. It then follows from this observation, the fact that RPF-SFISTA is called with functions (f, h) and ϕ as in (34) and inputs $(z_0, \bar{M}_0) = (\vartheta_{k-1}, \bar{N}_k)$, Theorem 2.2, and part (a) that RPF-SFISTA outputs a quadruple $(w_k, u_k, \vartheta_k, \bar{N}_k) = (y, v, \xi, L)$ that satisfies

$$\psi_{\delta_{k-1}}(\vartheta_k; \vartheta_{k-1}) \leq \min\left\{\psi_{\delta_{k-1}}(\vartheta_{k-1}; \vartheta_{k-1}), \psi_{\delta_{k-1}}(w_k; \vartheta_{k-1})\right\}, \quad \underline{\mathrm{N}}_k \leq \bar{N}_k \leq \max\left\{\underline{\mathrm{N}}_k, \kappa(\bar{L}_{\psi_s} + \delta_0)\right\}.$$

The relations in (38) then follow from the above relations, the fact that the way $\underline{\mathrm{N}}_k$ is chosen in step 1 implies that $\underline{\mathrm{N}}_k \geq \bar{N}_0$, the definition of $\bar{L}_{\delta_0}$ in (36), and the fact that the definition of ψ_δ in (37) implies that $\psi(\vartheta_k) \leq \psi_{\delta_{k-1}}(\vartheta_k; \vartheta_{k-1})$ and $\psi_{\delta_{k-1}}(\vartheta_{k-1}; \vartheta_{k-1}) = \psi(\vartheta_{k-1})$.

It also follows immediately from Theorem 2.2, the definition of ϵ-optimal solution, and the fact that RPF-SFISTA is called with tolerance $\hat{\epsilon} = \epsilon/6$ and function pair (f, h) as in (34) that pair $(w_k, u_k) = (y, v)$ satisfies both relations in (39).

(c) Consider the call made to RPF-SFISTA during the k-th iteration of A-REG. It follows directly from part (a), the fact that $\kappa \geq 1$, and the definition of $\bar{L}_{\delta_0}$ in (36) that the function f in (34) is $\bar{L}_{\delta_0}$-smooth and hence satisfies relation (4) with $\bar{L} = \bar{L}_{\delta_0}$. Part (a) implies that ϕ is δ_{k-1}-strongly

convex and thus satisfies assumption (A3) with $\bar{\mu} = \delta_{k-1}$. It also follows from the definitions of ϕ and ψ_δ in (34) and (37), respectively that $\phi(\vartheta_{k-1}) = \psi(\vartheta_{k-1})$ and that $\psi(w^*) \leq \min_x \psi_{\delta_{k-1}}(x; \vartheta_{k-1})$ where w^* is an optimal solution of (30). It then follows from these conclusions, the fact that the call to RPF-SFISTA is made with tolerance $\hat{\epsilon} = \epsilon/6$, functions (f, h) and ϕ as in (34), initial point $z_0 = \vartheta_{k-1}$, and inputs $(\mu_0, \bar{M}_0) = (B\delta_{k-1}, \underline{\mathrm{N}}_k)$, and the definitions of $C_{\bar{\mu}}(\cdot)$ and $\mathcal{C}_\delta(\cdot)$ in (16) and (37), respectively that RPF-SFISTA performs at most

$$\mathcal{O}_1 \left(\left\lceil \log_1^+ (2B) \right\rceil \left[\sqrt{\frac{2 \max\left\{\underline{\mathrm{N}}_k, \bar{L}_{\delta_0}\right\}}{\min\left\{B\delta_{k-1}, \delta_{k-1}/2\right\}}} \log_1^+ \left(\frac{36\left(\bar{L}_{\delta_0}^2 + \underline{\mathrm{N}}_k^2\right) \mathcal{C}_{\delta_{k-1}}(\vartheta_{k-1})}{\epsilon^2} \right) + \log(\bar{L}_{\delta_0}/\underline{\mathrm{N}}_k) \right] \right)$$

ACG iterations/resolvent evaluations. The result then follows from this conclusion, the fact that the way $\underline{\mathrm{N}}_k$ is chosen in step 1 of A-REG implies that $\underline{\mathrm{N}}_k \geq \bar{N}_0$, the fact that $B \geq 1$, and part (b). □

Lemma 3.3. *For every iteration index $k \geq 1$ generated by A-REG, the quantity $\underline{\mathrm{N}}_k$ satisfies*

$$\underline{\mathrm{N}}_k \leq \max\left\{\bar{N}_0, \bar{L}_{\delta_0}\right\} \tag{40}$$

where $\bar{N}_0$ is an input to A-REG and $\bar{L}_{\delta_0}$ is as in (36).

Proof. The proof follows from an induction argument. The fact that $\underline{\mathrm{N}}_1 = \bar{N}_0$ immediately implies that relation (40) holds for $k = 1$. Suppose now that $k \geq 2$ is an iteration index generated by A-REG and that inequality (40) holds for $k - 1$. It follows from the way $\underline{\mathrm{N}}_k$ is chosen in step 1 of the k-th iteration of A-REG and the first inequality in the last relation in (38) that $\underline{\mathrm{N}}_k \leq \bar{N}_{k-1}$. This relation and the last inequality in (38) with $k = k - 1$ then imply that

$$\underline{\mathrm{N}}_k \leq \bar{N}_{k-1} \overset{(38)}{\leq} \max\{\underline{\mathrm{N}}_{k-1}, \bar{L}_{\delta_0}\} \leq \max\left\{\bar{N}_0, \bar{L}_{\delta_0}\right\}$$

where the last inequality is due to the induction hypothesis. Hence, relation (40) holds for every iteration index $k \geq 1$ generated by A-REG. □

A useful fact that is used in the proof of following result is that if a function $\Psi : \mathbb{E} \to \Re \cup \{+\infty\}$ is ν-convex with modulus $\nu > 0$, then it has an unique global minimum x^* and

$$\Psi(x^*) + \frac{\nu}{2}\|\cdot - x^*\|^2 \leq \Psi(\cdot). \tag{41}$$

Lemma 3.4. *For any iteration index $k \geq 1$ generated by A-REG, the following relations hold*

$$r_k \in \nabla\psi_s(w_k) + \partial\psi_n(w_k) \tag{42}$$

$$\|\vartheta_k\| \leq D, \quad \|w_k\| \leq D + \frac{\epsilon}{3\delta_{k-1}} \tag{43}$$

where $D > 0$ is as in Assumption 3.1 and $\epsilon > 0$ is the input tolerance to A-REG.

Proof. Relation (42) follows immediately from the inclusion in (39) and the definition of r_k in (35). It follows immediately from the first inequality in (38) and a simple induction argument that $\psi(\vartheta_k) \leq \psi(\vartheta_0)$ for any iteration index $k \geq 1$ generated by A-REG. This relation and Assumption 3.1 then immediately imply that the first relation in (43) holds.

The second relation in (43) clearly holds if $\|w_k - \vartheta_k\| = 0$ since this relation together with the first relation in (43) implies that $\|w_k\| = \|\vartheta_k\| \leq D$. Hence, assume that $\|w_k - \vartheta_k\| > 0$. It follows

from the definition of $\psi_\delta(w;\vartheta)$ in (37) and the facts that ψ_s and ψ_n are convex that the inclusion in (39) is equivalent to $u_k \in \partial\psi_{\delta_{k-1}}(w_k;\vartheta_{k-1})$. This inclusion implies that

$$w_k = \operatorname*{argmin}_x \psi_{\delta_{k-1}}(x;\vartheta_{k-1}) - \langle u_k, x - w_k\rangle.$$

It then follows from this relation, the fact that the definition of $\psi_\delta(w;\vartheta)$ in (37) implies that $\psi_{\delta_{k-1}}(x;\vartheta_{k-1}) - \langle u_k, x - w_k\rangle$ is δ_{k-1}-strongly convex, and relation (41) with $\Psi(\cdot) = \psi_{\delta_{k-1}}(\cdot;\vartheta_{k-1}) - \langle u_k, \cdot - w_k\rangle$, $x^* = w_k$, and $\nu = \delta_{k-1}$ that

$$\psi_{\delta_{k-1}}(w_k;\vartheta_{k-1}) + \frac{\delta_{k-1}}{2}\|x - w_k\|^2 \le \psi_{\delta_{k-1}}(x;\vartheta_{k-1}) - \langle u_k, x - w_k\rangle \tag{44}$$

for all $x \in \mathbb{E}$. Relation (44) with $x = \vartheta_k$ and the second relation in (38) imply that $\langle u_k, w_k - \vartheta_k\rangle \ge 0.5\delta_{k-1}\|\vartheta_k - w_k\|^2$ which together with the fact that $\|w_k - \vartheta_k\| > 0$ and Cauchy-Schwarz inequality implies that $0.5\delta_{k-1}\|\vartheta_k - w_k\| \le \|u_k\|$. This relation, the first relation in (43), the inequality in (39), and reverse triangle inequality then imply that

$$\|w_k\| \le \|\vartheta_k\| + \|\vartheta_k - w_k\| \overset{(43)}{\le} D + \frac{2}{\delta_{k-1}}\|u_k\| \overset{(39)}{\le} D + \frac{\epsilon}{3\delta_{k-1}}$$

from which the second relation in (43) immediately follows. □

The following proposition establishes an upper bound on the number of iterations that A-REG performs and, consequently, a lower bound on the quantity δ_k.

Proposition 3.5. *A-REG performs at most* $\lceil \log_1^+(8D\delta_0/\epsilon)\rceil$ *iterations to find a pair* (w, r) *that satisfies* (32). *Moreover, for every iteration index* $k \ge 1$ *generated by A-REG, it holds that* $\delta_{k-1} \ge \min\{\delta_0, \epsilon/(8D)\}$.

Proof. In view of the fact that the first iteration of A-REG is performed with δ_0, the way δ_k is updated at the end of step 3 of A-REG, and the fact that for any iteration index k generated by A-REG pair (w_k, r_k) always satisfies inclusion (42), to show both conclusions of the proposition it suffices to show that if the k-th iteration of A-REG is performed with $\delta_{k-1} \le \epsilon/(4D)$, then $\|r_k\| \le \epsilon$ and hence A-REG terminates in its k-th iteration.

Hence, assume that the k-th iteration of A-REG is performed with $\delta_{k-1} \le \epsilon/(4D)$. It then follows from this relation, the definition of r_k in (35), triangle inequality, the inequality in (39), and both relations in (43) that

$$\begin{aligned}\|r_k\| &\overset{(35)}{\le} \|u_k\| + \delta_{k-1}\|\vartheta_{k-1} - w_k\| \overset{(39)}{\le} \frac{\epsilon}{6} + \delta_{k-1}\|\vartheta_{k-1} - w_k\| \\ &\le \frac{\epsilon}{6} + \delta_{k-1}\left(\|\vartheta_{k-1}\| + \|w_k\|\right) \overset{(43)}{\le} \frac{\epsilon}{6} + \delta_{k-1}\left(2D + \frac{\epsilon}{3\delta_{k-1}}\right) = \frac{\epsilon}{2} + 2D\delta_{k-1} \le \epsilon.\end{aligned}$$

It then follows from the above relation that A-REG terminates in step 3 of its k-th iteration with a pair $(w, r) = (w_k, r_k)$ satisfying (32) and hence both conclusions of the proposition hold. □

We are now ready to prove Theorem 3.1.

Proof. It follows from the first relation in (38) and an induction argument that $\psi(\vartheta_k) \le \psi(\vartheta_0)$ for any iteration index $k \ge 1$ generated by A-REG. It then follows from this fact and the definitions of ψ^0 and $\mathcal{C}_\delta(\cdot)$ in (36) and (37), respectively, that $36\mathcal{C}_{\delta_{k-1}}(\vartheta_{k-1}) \le (288\psi^0)/\delta_{k-1}$. It then follows

from this relation, the definition of $\bar{\mathcal{L}}^2$ in (36), the bounds on $\underline{\mathrm{N}}_k$ and δ_k in (40) and Proposition 3.5, respectively, and Lemma 3.2(c) that the call made to the RPF-SFISTA method during the k-th iteration of A-REG performs at most

$$\left(\lceil \log_1^+ (8D\delta_0/\epsilon) \rceil \lceil \log_1^+ 2B \rceil \left[\sqrt{\frac{2\max\left\{\bar{N}_0, \bar{L}_{\delta_0}\right\}}{\min\left\{\delta_0, \epsilon/(8D)\right\}}} \log_1^+ \left(\frac{288\psi^0 \bar{\mathcal{L}}^2}{\min\left\{\delta_0, \epsilon/(8D)\right\}\epsilon^2} \right) + \log(\bar{L}_{\delta_0}/\bar{N}_0) \right] \right)$$

ACG iterations/resolvent evaluations. The result then immediately follows from this observation and the first conclusion of Proposition 3.5. □

4 Numerical Experiments

This section benchmarks the numerical performance of RPF-SFISTA against four other state-of-the-art methods for solving four classes of strongly convex/convex composite optimization problems. It contains five subsections. The first subsection reports the numerical performance of all 5 methods on sparse logistic regression problems, the second one reports the performance of the methods on Lasso problems, while the third and fourth subsections report the numerical performance of all the methods on dense vector quadratic programs constrained to the simplex and box, respectively. The last subsection contains comments about the numerical results.

We now describe our implementation of RPF-SFISTA. First, the input parameters of RPF-SFISTA are chosen as

$$\beta = 1.25, \quad \chi = 0.001, \quad \underline{\mathrm{M}}_1 = 10.$$

Second, for each problem instance, the initial strong convexity estimate μ_0 is always taken as

$$\mu_0 = \frac{4\left[f(y_1) - \ell_f(y_1; \tilde{x}_0)\right]}{(1-\chi)\|y_1 - \tilde{x}_0\|^2}$$

where y_1 and $\tilde{x}_0$ are generated at the end of step 3 of the first cycle of RPF-SFISTA. Third, each time RPF-SFISTA restarts, the next strong convexity estimate μ_l is taken to be $\mu_l = 0.1\mu_{l-1}$. Finally, for $l \geq 2$, $\underline{\mathrm{M}}_l$ is chosen as $0.4\bar{M}_{l-1}$.

RPF-SFISTA is bench-marked against the following four state-of-the-art algorithms. Specifically, we consider the FISTA method with backtracking presented in [6] (nicknamed FISTA-BT), the restarted method of [33] (nicknamed FISTA-R), the RADA-FISTA method of [23] (nicknamed RA-FISTA), and the Greedy-FISTA method of [23] (nicknamed GR-FISTA). FISTA-R restarts based on a heuristic function value restarted scheme. RADA-FISTA and Greedy-FISTA are restarted FISTA methods that restart based on the heuristic gradient restarted scheme presented in [33].

Next, we describe the implementation details of the four algorithms which we compare RPF-SFISTA with. The implementation of FISTA-BT adaptively searches for a Lipschitz estimate L_j by a procedure similar to step 2 of RPF-SFISTA. Specifically, it checks an inequality similar to (8) with $\chi = 0.001$ and if the inequality does not hold, it doubles its Lipschitz estimate L_j, regenerates its potential next iterate y_j, and checks the inequality again. Also, like RPF-SFISTA, FISTA-BT sets 10 as its initial guess for the Lipschitz constant. FISTA-R restarts FISTA-BT (as described above) if FISTA-BT finds a point y_j with worse objective value than the previous point, i.e., if $\phi(y_j) > \phi(y_{j-1})$. It also sets 10 as an initial estimate for the Lipchitz constant. The implementations of RADA-FISTA and Greedy-FISTA are taken directly from the authors' Github repository, https://github.com/jliang993/Faster-FISTA. The input parameters of RADA-FISTA are chosen as

$$p = 0.5, \quad q = 0.5, \quad r = 4, \quad \gamma = \frac{1}{\bar{L}}$$

where $\bar{L}$ is the global Lipschitz constant of the gradient of f. All input parameters are chosen so as to meet the required ranges presented in Algorithm 4.2 of [23]. Greedy-FISTA only takes in a single input, a stepsize γ, which is chosen as $1.3/\bar{L}$. This more aggressive choice of stepsize is in the range suggested in the paragraph following Algorithm 4.3 in [23] and is the stepsize used in several of the experiments presented in the authors' Github repository. The code further has a safeguard which allows the stepsize to be decreased if a certain condition is satisfied.

We now describe the type of solution each of the methods aims to find. That is, given functions f and h satisfying assumptions (B1)-(B3) described in Section 2, an initial point $z_0 \in \mathcal{H}$, and tolerance $\hat{\epsilon} > 0$, each of the methods aims to find a pair (z, v) satisfying:

$$v \in \nabla f(z) + \partial h(z), \quad \frac{\|v\|}{1 + \|\nabla f(z_0)\|} \leq \hat{\epsilon} \tag{45}$$

where $\|\cdot\|$ signifies the Euclidean norm.

The tables below report the runtimes and the total number of ACG iterations/resolvent evaluations needed to find a pair (z, v) satisfying (45). A tolerance of either 10^{-8} or 10^{-13} is set and a time limit of 7200 seconds (2 hours) is given. An entry of a table marked with $*/N$ means that the corresponding method finds an approximate solution with relative accuracy strictly larger than the desired accuracy in which case N expresses the minimum relative accuracy that the method achieved within the time limit of 2 hours. The bold numbers in the tables of this section indicate the algorithm that performed the best for that particular metric (i.e. runtime or ACG iterations).

It will be seen from the numerical results presented in Subsections 4.1, 4.2, 4.3, and 4.4 that our method, RPF-SFISTA, was the fastest method and the only method able to find a solution of the desired accuracy (either 10^{-8} or 10^{-13}) on every instance considered. To compare RPF-SFISTA and the second-best performing method on a particular problem class more closely, we also report in each table caption the following average time ratio (ATR) between the two methods defined as

$$ATR = \frac{1}{N} \sum_{i=1}^{N} b_i / r_i, \tag{46}$$

where N is the number of class instances that both methods were tested on and b_i and r_i are the runtimes of the second-best performing method and RPF-SFISTA for instance i, respectively. If a method did not finish within the time limit of 7200 seconds on a particular instance, then its runtime for that instance is conservatively recorded as 7200 seconds in the computation of the ATR metric.

All experiments were performed in MATLAB 2024a and run on a Macbook Pro with a Apple M3 Max chip and 128 GB of memory.

4.1 Sparse logistic regression

Consider the problem

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \left[f(z) := \sum_{i=1}^{m} \log\left(1 + \exp(-b_i \langle a_i, z \rangle)\right) \right] \\ \text{s.t.} \quad & \|z\|_1 \leq C \end{aligned}$$

where $a_i \in \mathbb{R}^n$ and $b_i \in \{-1, 1\}$ for $i \in [m]$, and $C > 0$ is a regularization parameter. For our experiments in § 4.1, we vary the dimension pair (m, n) and C is taken to be either 0, 0.5, or 1. For each instance, the initial point x_0 is chosen to be a random point satisfying $\|x_0\|_1 \leq C$. The Lipschitz constant $\bar{L}$ of f is upper bounded by $0.25\lambda_{\max}(D^T D)$ where $D \in \mathbb{R}^{m \times n}$ satisfies $D_{ij} = -a_i^j b_i$ and a_i^j denotes the j-th entry of a_i.

Parameters	Lipschitz	Iteration Count/Runtime (seconds)				
(m,n,C)	$\bar{L}$	RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(500, 50,000, 0.5)	$1.56 * 10^6$	**429/32.75**	12140/1013.03	5357/529.51	6483/463.16	6303/410.17
(500, 50,000, 1)	$1.56 * 10^6$	**802/62.23**	85066/6958.55	21177/2065.41	7069/490.30	6474/410.03
(500, 50,000, 2)	$1.56 * 10^6$	**909/71.02**	59479/4766.41	19469/1886.22	8445/589.76	7889/508.38
(1000, 250,000, 0.5)	$1.56 * 10^7$	**1110/739.98**	*/6.51e-07	*/3.27e-08	*/4.91e-08	14898/6694.58
(1000, 250,000, 1)	$1.56 * 10^7$	**1375/943.30**	*/2.67e-06	*/8.08e-07	*/2.30e-07	*/1.52e-07
(1000, 250,000, 2)	$1.56 * 10^7$	**1712/1199.78**	*/2.71e-06	*/1.33e-07	*/2.38e-07	*/2.25e-07
(300, 500,000, 0.5)	$9.39 * 10^6$	**598/230.24**	13383/5313.30	4355/2224.32	23569/7087.33	19019/5807.34
(300, 500,000, 1)	$9.39 * 10^6$	**1421/561.68**	*/8.47e-08	*/8.18e-08	*/1.47e-07	23125/6971.15
(300, 500,000, 2)	$9.39 * 10^6$	**977/382.66**	*/1.08e-06	*/4.47e-07	*/2.64e-07	*/7.62e-08
(100, 1,000,000, 0.5)	$6.27 * 10^6$	**588/178.35**	6129/1938.83	2824/1126.19	*/4.70e-08	24554/6419.97
(100, 1,000,000, 1)	$6.27 * 10^6$	**1389/431.63**	*/7.74e-08	8845/3530.80	*/5.50e-07	*/2.09e-07
(100, 1,000,000, 2)	$6.27 * 10^6$	**2099/673.07**	*/1.93e-06	*/4.92e-07	*/5.29e-06	*/3.98e-06

Table 4: Iteration counts and runtimes (in seconds) for the sparse logistic regression problem in § 4.1. The tolerances are set to 10^{-8}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 14.06.

4.2 Lasso

Consider the problem

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \ & \left[f(z) := \frac{1}{2}\|Az - b\|^2 \right] \\ \text{s.t. } & \|z\|_1 \leq C \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ and $C > 0$ is a regularization parameter. For our experiments in § 4.2, the matrix A and the vector b are taken from the linear programming test problems considered in the datasets 'Meszaros' and 'Mittelmann'. The regularization parameter, C is taken to be either 1, 5, or 10 and for each problem instance, the initial point x_0 is chosen to be a random point satisfying $\|x_0\|_1 \leq C$. The Lipschitz constant $\bar{L}$ of the gradient of f is just $\|A\|^2$. The results comparing the five methods are presented in Tables 5 and 6, where the names of each problem instance considered from the Meszaros and Mittelmann datasets are also given.

4.3 Dense QP with Simplex Constraints

Given a pair of dimensions $(m, n) \in \mathbb{N}^2$, a scalar pair $(\tau_1, \tau_2) \in \Re^2_{++}$, matrices $B \in \Re^{n \times n}$ and $C \in \Re^{m \times n}$, positive diagonal matrix $D \in \Re^{n \times n}$, and a vector $d \in \Re^m$, this subsection considers the problem

$$\begin{aligned} \min_{z} \ & \left[f(z) := \frac{\tau_1}{2}\|DBz\|^2 + \frac{\tau_2}{2}\|Cz - d\|^2 \right] \\ \text{s.t. } & z \in \Delta^n, \end{aligned}$$

where $\Delta^n := \{x \in \Re^n_+ : \sum_{i=1}^n x_i = 1\}$. For our experiments in § 4.3, we vary the dimensions and generate the matrices B and C to be fully dense. The entries of B, C, and d (resp. D) are generated by sampling from the uniform distribution $\mathcal{U}[0, 1]$ (resp. $\mathcal{U}[1, \alpha]$) where the parameter $\alpha \geq 1$ is varied. The initial starting point x_0 is generated as $\hat{x}/\sum_{i=1}^n \hat{x}_i$, where the entries of $\hat{x}$ are sampled from the $\mathcal{U}[0, 1]$ distribution. Finally, we choose $(\tau_1, \tau_2) \in \Re^2_{++}$ so that $\bar{L} = \lambda_{\max}(\nabla^2 f)$ and $\bar{\mu} = \lambda_{\min}(\nabla^2 f)$ are the various values given in the tables of this subsection. The results comparing

Parameters	Lipschitz	Name	Iteration Count/Runtime (seconds)				
(m,n,C)	$\bar{L}$		RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(825, 8,627, 1)	$1.55*10^3$	aa03	**226/0.28**	4888/4.86	4598/4.45	706/0.86	636/0.66
(825, 8,627, 5)	$1.55*10^3$	aa03	**722/0.72**	8482/8.07	8482/7.82	1322/1.44	1198/1.06
(825, 8,627, 10)	$1.55*10^3$	aa03	**472/0.52**	13132/13.04	9654/9.18	1352/1.47	1239/1.14
(426, 7,195, 1)	$1.73*10^3$	aa4	**348/0.29**	1522/1.27	1461/1.22	833/0.83	735/0.56
(426, 7,195, 5)	$1.73*10^3$	aa4	**432/0.44**	6898/5.87	6898/5.48	1589/1.44	1605/1.30
(426, 7,195, 10)	$1.73*10^3$	aa4	**735/0.64**	10377/10.03	8503/7.39	2289/2.14	2203/1.75
(50, 6,774, 1)	$1.73*10^4$	air02	**370/0.34**	5223/4.32	3570/2.66	3482/3.32	3380/2.65
(50, 6,774, 5)	$1.73*10^4$	air02	**3410/2.78**	125979/102.89	124619/87.70	28156/23.63	21579/15.82
(50, 6,774, 10)	$1.73*10^4$	air02	**5078/2.59**	129536/57.52	131588/61.23	1551/1.07	1353/0.59
(124, 10,757, 1)	$1.57*10^4$	air03	**627/11.80**	7138/127.32	5173/78.53	3211/42.12	3488/65.76
(124, 10,757, 5)	$1.57*10^4$	air03	**1090/17.61**	41501/618.35	41501/605.35	9257/139.24	10359/219.91
(124, 10,757, 10)	$1.57*10^4$	air03	**2996/70.14**	286189/4624.86	289853/4549.27	93958/1598.77	30278/552.48
(823, 8,904, 1)	$1.50*10^3$	air04	**339/0.32**	5233/5.20	3139/3.13	1150/1.20	980/1.02
(823, 8,904, 5)	$1.50*10^3$	air04	**342/0.36**	5575/5.59	5085/5.35	1163/1.26	977/1.01
(823, 8,904, 10)	$1.50*10^3$	air04	**903/0.86**	10956/10.02	9270/8.62	1257/1.39	1342/1.36
(769, 2,561, 1)	$1.54*10^3$	gen	**93/0.06**	450/0.19	323/0.15	331/0.15	254/0.11
(769, 2,561, 5)	$1.54*10^3$	gen	**177/0.09**	1488/0.66	948/0.43	498/0.20	454/0.18
(769, 2,561, 10)	$1.54*10^3$	gen	**190/0.11**	2613/1.09	1491/0.69	689/0.30	591/0.27
(163, 28,016, 1)	$5.64*10^4$	us04	**438/7.57**	11162/158.37	9858/135.20	5954/99.42	6357/104.38
(163, 28,016, 5)	$5.64*10^4$	us04	**1119/21.39**	41307/644.84	33990/520.40	8530/135.05	9609/171.61
(163, 28,016, 10)	$5.64*10^4$	us04	**1713/33.49**	422708/6284.62	446469/6230.74	14290/215.23	14280/205.82
(520, 1,544, 1)	$3.10*10^4$	rosen1	30/0.009	30/0.007	30/0.006	31/0.007	**14/0.005**
(520, 1,544, 5)	$3.10*10^4$	rosen1	**87/0.02**	151/0.02	128/0.02	151/0.03	108/0.02
(520, 1,544, 10)	$3.10*10^4$	rosen1	**101/0.02**	288/0.05	205/0.04	231/0.05	181/0.03
(2,056, 6,152, 1)	$1.17*10^5$	rosen10	29/0.02	29/0.02	29/0.02	58/0.04	**20/0.02**
(2,056, 6,152, 5)	$1.17*10^5$	rosen10	**163/0.13**	624/0.39	363/0.22	449/0.31	388/0.26
(2,056, 6,152, 10)	$1.17*10^5$	rosen10	**209/0.16**	2491/1.62	1172/0.78	535/0.40	421/0.25
(135, 6,469, 1)	$4.93*10^3$	crew1	**477/0.28**	9519/5.31	8573/5.12	2689/1.57	2821/1.67
(135, 6,469, 5)	$4.93*10^3$	crew1	**1790/1.25**	44385/34.44	31153/26.13	4655/3.65	5529/4.11
(135, 6,469, 10)	$4.93*10^3$	crew1	**2362/1.63**	1000023/718.28	1000023/764.45	9093/6.88	9473/6.92

Table 5: Iteration counts and runtimes (in seconds) for the Lasso problem in § 4.2. The tolerances are set to 10^{-13}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 3.87.

the five methods are presented in Tables 7 and 8. Table 7 and Table 8 present the performance of all five methods on the same exact 12 instances, but for target accuracies $\epsilon = 10^{-8}$ and $\epsilon = 10^{-13}$, respectively.

4.4 Dense QP with Box Constraints

Given a pair of dimensions $(m, n) \in \mathbb{N}^2$, a scalar triple $(r, \tau_1, \tau_2) \in \Re^3_{++}$, matrices $B \in \Re^{n\times n}$ and $C \in \Re^{m\times n}$, positive diagonal matrix $D \in \Re^{n\times n}$, a vector pair $(a, d) \in \Re^n \times \Re^m$, and a scalar $b \in \Re$, this subsection considers the problem

$$\begin{aligned} \min_z \ & \left[f(z) := \frac{\tau_1}{2}\|DBz\|^2 + \frac{\tau_2}{2}\|Cz - d\|^2 \right] \\ \text{s.t. } & a^T z = b \end{aligned}$$

Parameters	**Lipschitz**	**Name**	**Iteration Count/Runtime (seconds)**				
(m,n,C)	$\bar{L}$		RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(905, 1,513, 1)	$2.14*10^5$	cr42	161/0.05	451/0.07	408/0.07	81/0.01	**83/0.01**
(905, 1,513, 5)	$2.14*10^5$	cr42	765/0.15	11628/1.89	4503/0.74	627/0.09	**395/0.06**
(905, 1,513, 10)	$2.14*10^5$	cr42	8180/1.47	197048/39.93	39202/8.13	4027/0.77	**3465/0.62**
(71, 36,699, 1)	$2.49*10^4$	kl02	**2585/75.80**	8633/188.86	8633/192.25	5812/139.33	5966/138.34
(71, 36,699, 5)	$2.49*10^4$	kl02	659/21.54	1574/39.99	*/2.99e-07	459/10.27	**412/8.59**
(71, 36,699, 10)	$2.49*10^4$	kl02	500/11.22	1571/27.56	811/22.23	374/6.49	**345/6.30**
(664, 46,915, 1)	$1.58*10^4$	t0331-4l	**499/10.67**	4742/94.66	3291/85.86	1988/43.69	1963/35.04
(664, 46,915, 5)	$1.58*10^4$	t0331-4l	**854/20.36**	22202/440.41	15215/391.61	3726/87.56	4589/80.62
(664, 46,915, 10)	$1.58*10^4$	t0331-4l	**598/13.81**	115672/2826.81	91697/2846.42	3959/82.91	3846/77.28
(507, 63,516, 1)	$2.23*10^4$	rail507	**441/9.45**	16380/322.10	14546/379.22	3871/89.02	3944/79.39
(507, 63,516, 5)	$2.23*10^4$	rail507	**633/16.98**	33798/946.03	25542/801.14	5445/150.25	5724/155.04
(507, 63,516, 10)	$2.23*10^4$	rail507	**1081/26.66**	54412/1198.24	42622/1202.48	6682/167.30	7649/179.36
(516, 47,827, 1)	$2.07*10^4$	rail516	**1132/24.25**	5795/96.74	5795/131.08	3890/76.33	3719/91.63
(516, 47,827, 5)	$2.07*10^4$	rail516	**615/14.33**	16843/318.72	12423/295.14	5053/129.64	5188/154.17
(516, 47,827, 10)	$2.07*10^4$	rail516	**1053/26.65**	25057/563.01	25057/720.52	6155/163.21	5688/155.14
(582, 56,097, 1)	$3.46*10^4$	rail582	**656/16.51**	10000/263.01	10000/327.45	7427/196.50	7139/218.76
(582, 56,097, 5)	$3.46*10^4$	rail582	**518/14.02**	*/4.06e-12	*/5.35e-12	7943/210.25	7978/251.48
(582, 56,097, 10)	$3.46*10^4$	rail582	**1255/36.54**	*/4.23e-11	*/2.60e-11	8036/195.24	8278/248.95
(2,586, 923,269, 1)	$2.46*10^5$	rail2586	**3254/343.08**	*/5.98e-11	*/8.33e-11	24725/2631.23	25618/2729.25
(2,586, 923,269, 5)	$2.46*10^5$	rail2586	**2039/203.52**	*/2.21e-10	*/2.42e-10	*/6.14e-11	26307/2614.78
(2,586, 923,269, 10)	$2.46*10^5$	rail2586	**1843/163.57**	*/1.07e-09	*/1.22e-09	26475/2630.74	31177/3194.57
(4,284, 1,096,894, 1)	$1.60*10^5$	rail4284	**4138/523.35**	*/1.61e-11	*/1.62e-11	30633/3974.30	30391/3909.32
(4,284, 1,096,894, 5)	$1.60*10^5$	rail4284	**3852/454.55**	*/5.05e-11	*/5.66e-11	*/7.70e-08	*/7.53e-08
(4,284, 1,096,894, 10)	$1.60*10^5$	rail4284	**2970/340.49**	*/1.06e-09	*/1.15e-09	*/6.86e-11	31325/3394.41
(73, 123,409, 1)	$2.56*10^5$	nw14	**793/28.11**	4828/167.18	3416/114.27	4718/202.63	4527/178.69
(73, 123,409, 5)	$2.56*10^5$	nw14	**4753/186.80**	96469/3317.70	80492/2513.14	10451/341.44	9860/310.00
(73, 123,409, 10)	$2.56*10^5$	nw14	**2460/92.08**	120336/3364.65	86166/2366.60	18593/604.72	18294/593.66
(27,441, 30,733, 1)	$3.6*10^{11}$	baxter	**2431/35.26**	4671/70.73	4276/65.40	5098/67.13	4339/44.83
(27,441, 30,733, 5)	$3.6*10^{11}$	baxter	11805/140.86	12352/186.97	9150/134.82	11976/152.81	**8329/82.57**
(27,441, 30,733, 10)	$3.6*10^{11}$	baxter	20929/240.32	21569/322.11	19936/307.04	16833/210.20	**12231/122.60**

Table 6: Iteration counts and runtimes (in seconds) for the Lasso Problem in § 4.2. The tolerances are set to 10^{-13}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 6.32.

$$-r \leq z_i \leq r, \quad i \in \{1, ..., n\}.$$

For our experiments in § 4.4, we vary the dimensions (m, n) and generate the matrices B and C to be fully dense. The entries of B, C, and d (resp. D) are generated by sampling from the uniform distribution $\mathcal{U}[0, 1]$ (resp. $\mathcal{U}[1, 1000]$). The vector a is varied to be either the vector that takes value -1 in its last component and value 1 in all its other components or the vector that takes value -1 in its last 10 components and value 1 in all its other components. The scalars b and r are taken to be 0 and 5, respectively. The initial starting point x_0 is generated as a random vector in $\mathcal{U}[-r, r]^n$. Finally, we choose $(\tau_1, \tau_2) \in \Re^2_{++}$ so that $\bar{L} = \lambda_{\max}(\nabla^2 f)$ and $\bar{\mu} = \lambda_{\min}(\nabla^2 f)$ are the various values given in the tables of this subsection. Table 9 and Table 10 present the performance of all five methods on the same exact 12 instances, but for target accuracies $\epsilon = 10^{-8}$ and $\epsilon = 10^{-13}$, respectively.

Dimensions	**Curvatures**	**Iteration Count/Runtime (seconds)**				
(m,n)	$(\bar{\mu},\bar{L})$	RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(1000, 5000)	$(10^{-8},10^{2})$	**760/1261.82**	*/1.13e-07	*/8.91e-08	2385/3050.66	1929/2460.67
(1000, 5000)	$(10^{-6},10^{2})$	**322/536.82**	3201/4345.55	3201/4529.81	2291/2924.71	2163/2781.60
(1000, 5000)	$(10^{-4},10^{3})$	**140/224.95**	619/886.74	619/886.54	2090/2685.77	1714/2206.25
(1000, 5000)	$(10^{-6},10^{3})$	**113/181.25**	623/991.30	623/854.16	2569/3258.33	2593/3509.69
(1000, 5000)	$(10^{-7},10^{4})$	**88/141.95**	357/585.77	357/534.64	2840/3649.55	2719/3512.51
(1000, 5000)	$(10^{-4},10^{6})$	**72/108.84**	131/197.84	102/151.87	3231/4136.42	2735/3483.86
(2000, 10000)	$(10^{-4},10^{4})$	**120/1162.83**	249/2413.96	249/2549.80	*/7.41e-05	*/2.67e-05
(2000, 10000)	$(10^{-4},10^{4})$	**69/694.64**	128/1257.82	83/823.70	*/7.91e-05	*/2.77e-05
(2000, 10000)	$(10^{-4},10^{4})$	**75/735.75**	131/1287.77	97/965.95	*/3.20e-02	*/2.76e-03
(2000, 10000)	$(10^{-4},10^{6})$	**57/540.71**	89/820.29	74/716.82	*/3.93e-03	*/2.27e-03
(2000, 10000)	$(10^{-4},10^{3})$	**191/1937.20**	685/6795.16	685/6895.35	*/6.57e-05	*/2.93e-05
(2000, 10000)	$(10^{-4},10^{4})$	**92/941.71**	249/2406.88	156/1563.07	*/3.51e-03	*/1.14e-03

Table 7: Iteration counts and runtimes (in seconds) for the Vector QP with Simplex Constraints in § 4.3. The tolerances are set to 10^{-8}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 3.27.

Dimensions	**Curvatures**	**Iteration Count/Runtime (seconds)**				
(m,n)	$(\bar{\mu},\bar{L})$	RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(1000, 5000)	$(10^{-8},10^{2})$	**1335/1900.39**	*/7.91e-08	*/1.02e-07	4009/5030.54	3591/4570.78
(1000, 5000)	$(10^{-6},10^{2})$	**533/847.84**	*/6.08e-10	*/1.10e-09	4037/5120.40	3627/4674.44
(1000, 5000)	$(10^{-4},10^{3})$	**221/368.37**	1971/2643.61	1971/2807.98	3860/4904.65	3320/4256.82
(1000, 5000)	$(10^{-6},10^{3})$	**169/292.13**	1702/2445.07	1702/2530.27	4606/5894.16	4000/5128.65
(1000, 5000)	$(10^{-7},10^{4})$	**130/226.82**	902/1473.35	902/1431.87	4704/6004.52	4187/5341.55
(1000, 5000)	$(10^{-4},10^{6})$	**100/157.56**	274/457.90	248/377.36	4798/6149.02	4193/5376.22
(2000, 10000)	$(10^{-4},10^{4})$	**178/1808.51**	*/8.52e-13	*/9.83e-13	*/7.79e-05	*/2.53e-05
(2000, 10000)	$(10^{-4},10^{4})$	**133/1336.16**	394/3951.58	335/3365.69	*/2.34e-04	*/2.96e-05
(2000, 10000)	$(10^{-4},10^{4})$	**99/995.16**	310/3114.84	273/2759.90	*/4.56e-03	*/2.74e-03
(2000, 10000)	$(10^{-4},10^{6})$	**96/907.03**	191/1849.46	178/1705.75	*/4.13e-03	*/2.25e-03
(2000, 10000)	$(10^{-4},10^{3})$	**301/2996.59**	*/8.09e-09	*/6.79e-09	*/8.50e-04	*/2.78e-05
(2000, 10000)	$(10^{-4},10^{4})$	**142/1416.76**	*/3.29e-13	601/6022.82	*/2.34e-04	*/9.04e-05

Table 8: Iteration counts and runtimes (in seconds) for the Vector QP problem with Simplex Constraints in § 4.3. The tolerances are set to 10^{-13}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 4.59.

4.5 Comments about the numerical results

As seen from Tables 4, 5, 6, 7, 8, 9, and 10, RPF-SFISTA was the most efficient method and the only method able to find a solution of the desired accuracy (either 10^{-8} or 10^{-13}) within the time limit of 2 hours on every problem instance considered. Out of the 12 sparse logistic regression problems instances considered in § 4.1, both FISTA-BT and FISTA-R only finished within the time limit on 5 and 7 instances, respectively. Meanwhile, RA-FISTA and GR-FISTA finished on 4 and 7 instances, respectively. It can be seen from Table 4 that GR-FISTA was the second-best performing method while RPF-SFISTA was the best performing method for this problem class. As indicated by the ATR metric, RPF-SFISTA was on average over 14 times faster than GR-FISTA across all 12 instances.

Out of the 60 Lasso problem instances considered in § 4.2, both FISTA-BT and FISTA-R finished within the time limit on 52 and 51 instances, respectively. RA-FISTA and GR-FISTA

Dimensions	**Curvatures**	**Iteration Count/Runtime (seconds)**				
(m,n)	$(\bar{\mu},\bar{L})$	RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(500, 1000)	$(10^{-4}, 10^{2})$	**1035/62.34**	32483/1624.71	32483/1634.26	10568/443.47	7860/317.06
(500, 1000)	$(10^{-4}, 10^{2})$	**2922/165.28**	53356/2604.41	53356/2701.43	10619/441.68	7863/326.54
(500, 1000)	$(10^{-2}, 10^{4})$	**607/34.03**	8643/420.96	8643/436.37	9698/416.20	6820/284.80
(500, 1000)	$(10^{-2}, 10^{4})$	**1886/106.38**	48990/2392.33	48990/2475.34	12016/528.72	6847/285.02
(500, 1000)	$(10^{-3}, 10^{3})$	**900/51.24**	10023/489.64	10023/507.00	10584/461.42	8747/363.64
(500, 1000)	$(10^{-3}, 10^{3})$	**2948/166.56**	52019/2550.12	52019/1444.09	10640/450.38	7901/324.36
(1000, 2000)	$(10^{-2}, 10^{3})$	**577/113.21**	7507/1395.54	7507/1444.09	14361/2453.04	9767/1640.84
(1000, 2000)	$(10^{-2}, 10^{3})$	**2030/382.98**	*/1.37e-08	*/1.47e-08	14326/2452.20	9707/1633.46
(1000, 2000)	$(10^{-1}, 10^{4})$	**624/113.08**	7121/1317.06	7121/1367.63	14363/2456.39	9768/1642.85
(1000, 2000)	$(10^{-1}, 10^{4})$	**2054/376.27**	*/1.91e-08	*/2.10e-08	14327/2448.34	9708/1634.13
(1000, 2000)	$(10^{-1}, 10^{5})$	**586/105.60**	6919/1280.26	6919/1327.11	13962/2380.55	9940/1674.29
(1000, 2000)	$(10^{-1}, 10^{5})$	**2009/366.87**	*/1.94e-08	*/1.99e-08	13959/2385.47	9327/1572.13

Table 9: Iteration counts and runtimes (in seconds) for the Vector QP problem with Box Constraints in § 4.4. The tolerances are set to 10^{-8}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 7.08.

Dimensions	**Curvatures**	**Iteration Count/Runtime (seconds)**				
(m,n)	$(\bar{\mu},\bar{L})$	RPF-SFISTA	FISTA-BT	FISTA-R	RA-FISTA	GR-FISTA
(500, 1000)	$(10^{-4}, 10^{2})$	**1970/90.50**	*/1.81e-11	*/3.56e-11	17702/741.39	16033/677.15
(500, 1000)	$(10^{-4}, 10^{2})$	**6028/279.06**	*/3.33e-10	*/4.60e-10	18304/756.50	15284/632.51
(500, 1000)	$(10^{-2}, 10^{4})$	**1024/45.91**	42950/2096.65	42950/2180.87	15453/644.29	12308/518.62
(500, 1000)	$(10^{-2}, 10^{4})$	**3074/140.47**	*/1.32e-10	*/1.43e-10	17664/734.98	12472/525.10
(500, 1000)	$(10^{-3}, 10^{3})$	**1868/84.54**	62858/3063.96	62858/3183.97	18291/768.65	16425/685.00
(500, 1000)	$(10^{-3}, 10^{3})$	**6038/276.19**	*/2.34e-10	*/3.63e-10	18331/776.90	15297/641.28
(1000, 2000)	$(10^{-2}, 10^{3})$	**1132/210.81**	*/1.06e-12	*/9.77e-13	23314/3974.38	19020/3237.59
(1000, 2000)	$(10^{-2}, 10^{3})$	**3848/722.90**	*/1.45e-08	*/1.50e-08	24491/4187.51	18788/3199.92
(1000, 2000)	$(10^{-1}, 10^{4})$	**1200/222.01**	*/1.23e-12	*/8.72e-13	23293/3969.59	19033/3239.94
(1000, 2000)	$(10^{-1}, 10^{4})$	**4181/786.05**	*/2.09e-08	*/2.32e-08	24503/4173.13	18781/3196.47
(1000, 2000)	$(10^{-1}, 10^{5})$	**1104/201.44**	35820/6646.22	35820/6875.80	22498/3832.11	18868/3207.92
(1000, 2000)	$(10^{-1}, 10^{5})$	**3987/739.08**	*/1.88e-08	*/1.96e-08	22219/3782.47	19405/3300.10

Table 10: Iteration counts and runtimes (in seconds) for the Vector QP problem with Box Constraints in § 4.4. The tolerances are set to 10^{-13}. Entries marked with * did not converge in the time limit of 7200 seconds. The ATR metric is 7.84.

finished within the time limit of 2 hours on 57 and 59 instances, respectively. It can be seen from Tables 5 and 6 that GR-FISTA was the second-best performing method while RPF-SFISTA was the best performing method for this problem class. As indicated by the ATR metrics in both tables, RPF-SFISTA was approximately 3.87 times faster than GR-FISTA on the first 30 Lasso instances considered and 6.32 times faster on the last 30 instances considered. Hence, RPF-SFISTA was roughly 5 times faster than GR-FISTA across all 60 instances considered.

For the simplex-constrained dense QP instances considered in § 4.3, both FISTA-BT and FISTA-R finished with a solution of accuracy 10^{-8} within the time limit on 11 of the 12 instances while both RA-FISTA and GR-FISTA finished within the time limit on 6 instances. For this accuracy, FISTA-R was the second-best performing method on this problem class while RPF-SFISTA was the best performing method. As indicated by the ATR metric, RPF-SFISTA was on average over 3.3 times faster than FISTA-R across the 12 instances considered. When the desired accuracy was 10^{-13}, both FISTA-BT and FISTA-R finished within the time limit on 7 of the 12 instances

while both RA-FISTA and GR-FISTA finished on 6 instances. For this accuracy, FISTA-R was the second-best performing method while RPF-SFISTA was the best performing method. As indicated by the ATR metric, RPF-SFISTA was on average over 4.6 times faster than FISTA-R across the 12 instances, showing that the gap between RPF-SFISTA and the other codes increased when a higher accuracy of 10^{-13} was required.

For the box-constrained dense QP instances considered in § 4.4, both FISTA-BT and FISTA-R finished with a solution of accuracy 10^{-8} within the time limit of 2 hours on 9 of the 12 instances while both RA-FISTA and GR-FISTA finished on all 12 instances. For this accuracy, GR-FISTA was the second-best performing method on this problem class while RPF-SFISTA was the best performing method. As indicated by the ATR metric, RPF-SFISTA was on average over 7.08 times faster than GR-FISTA across the 12 instances considered. When the desired accuracy was 10^{-13}, both FISTA-BT and FISTA-R finished within the time limit on just 3 of the 12 instances while both RA-FISTA and GR-FISTA again finished on all 12 instances. GR-FISTA was the second-best performing method while RPF-SFISTA was the best performing method. As indicated by the ATR metric, RPF-SFISTA was on average over 7.8 times faster than GR-FISTA across the 12 instances.

A Proof of Proposition 2.1

This section is dedicated to proving Proposition 2.1. It is broken up into two subsections. The first subsection is dedicated to proving Proposition 2.1(a)-(b) while the second one is dedicated to proving Proposition 2.1(c).

A.1 Proof of Proposition 2.1(a)-(b)

The following lemma establishes a key bound on the distance between the iterate ξ_j and the initial point z_{l-1} of the l-th cycle.

Lemma A.1. *For every iteration index $j \geq 1$ generated during the l-th cycle of RPF-SFISTA, it holds that*

$$\|\xi_j - z_{l-1}\|^2 \leq C_{\bar{\mu}}(z_{l-1}) \tag{47}$$

where $C_{\bar{\mu}}(\cdot)$ is as in (16) and z_{l-1} is the initial point of the l-th cycle.

Proof. Let $j \geq 1$ be an iteration index generated during the l-th cycle of RPF-SFISTA. It follows immediately from the fact that ϕ is $\bar{\mu}$-convex and relation (41) with $\Psi = \phi$ and $\nu = \bar{\mu}$ that the following relations hold

$$\|\xi_j - z^*\|^2 \leq \frac{2}{\bar{\mu}}\left[\phi(\xi_j) - \phi(z^*)\right], \quad \|z_{l-1} - z^*\|^2 \leq \frac{2}{\bar{\mu}}\left[\phi(z_{l-1}) - \phi(z^*)\right]. \tag{48}$$

The above relations, triangle inequality, and relation (25) then imply that

$$\begin{aligned}\|\xi_j - z_{l-1}\|^2 &\leq 2\|\xi_j - z^*\|^2 + 2\|z^* - z_{l-1}\|^2 \\ &\overset{(48)}{\leq} \frac{4}{\bar{\mu}}\left[\phi(\xi_j) - \phi(z^*)\right] + \frac{4}{\bar{\mu}}\left[\phi(z_{l-1}) - \phi(z^*)\right] \overset{(25)}{\leq} \frac{8}{\bar{\mu}}\left[\phi(z_{l-1}) - \phi(z^*)\right].\end{aligned}$$

The conclusion of the lemma then immediately follows from the above relation and the definition of $C_{\bar{\mu}}(\cdot)$ in (16). □

The proof of Proposition 2.1(a)-(b) is now presented. The proof of part (a) has a similar pattern to the proof of Proposition A.5 in [38], but we include the proof here for the sake of completeness.

Proof of Proposition 2.1(a)-(b). (a) Consider the l-th cycle of RPF-SFISTA and let m denote the first quantity in (18). Using this definition and the inequality $\log(1+\alpha) \geq \alpha/(1+\alpha)$ for any $\alpha > -1$, it can easily be seen that

$$\left(1 + Q_l^{-1}\right)^{2(m-1)} \geq \frac{C_{\bar{\mu}}(z_{l-1})\zeta_l^2}{\chi \hat{\epsilon}^2}. \tag{49}$$

We claim that the l-th cycle of RPF-SFISTA terminates in either step 4 or step 5 in at most m iterations. It is thus sufficient to show that if the l-th cycle of RPF-SFISTA has not stopped in step 4 up to and including the m-th iteration, then it must stop successfully in step 5 at the m-th iteration. So, assume that the l-th cycle of RPF-SFISTA has not stopped in step 4 up to the m-th iteration. It is easy to see then in view of step 4 of RPF-SFISTA that relation (14) holds with $j = m$.

It then follows from this relation, inequality (23) with $j = m$, the fact that x_0 is set as z_{l-1} at the beginning of the l-th cycle, relations (47) and (49), and inequality (24) with $j = m$ that

$$C_{\bar{\mu}}(z_{l-1}) \overset{(47)}{\geq} \|\xi_m - z_{l-1}\|^2 = \|\xi_m - x_0\|^2 \overset{(14)}{\geq} \chi A_m L_m \|y_m - \tilde{x}_{m-1}\|^2 \overset{(23)}{\geq} \frac{\chi}{\zeta_l^2} A_m L_m \|v_m\|^2 \tag{50}$$

$$\overset{(24)}{\geq} \frac{\chi}{\zeta_l^2}\left(1 + Q_l^{-1}\right)^{2(m-1)} \|v_m\|^2 \overset{(49)}{\geq} \frac{C_{\bar{\mu}}(z_{l-1})}{\hat{\epsilon}^2}\|v_m\|^2 \tag{51}$$

which implies that the termination criterion (15) in step 5 of RPF-SFISTA is satisfied. Hence, the l-th cycle of RPF-SFISTA must successfully stop at the end of its m-th iteration and the claim thus holds. Moreover, it is easy to see from relation (22) that the second quantity in (18) is a bound on the total number of times that step 2 of RPF-SFISTA needs to be repeated. Since every time step 2 of RPF-SFISTA is performed only one resolvent evaluation is needed, the conclusion of part (a) follows.

(b) To show part (b), assume that the l-th cycle of RPF-SFISTA terminates in its step 5 and outputs a quadruple (y, v, ξ, L). It then follows that $(y, v, \xi, L) = (y_j, v_j, \xi_j, L_j)$ where j is an iteration index generated during the l-th cycle of RPF-SFISTA. The inclusion in (23) and the termination criterion (15) in step 5 of RPF-SFISTA then immediately imply that pair $(y, v) = (y_j, v_j)$ satisfies (5) and hence is an $\hat{\epsilon}$-optimal solution of (1). Moreover, it follows from the fact that $L = L_j$ for an iteration index j generated during the l-th cycle, both inequalities in (22), the fact that the way $\underline{\mathrm{M}}_l$ is chosen in step 0 implies that $\underline{\mathrm{M}}_l \geq \bar{M}_0$, and relation (26) that

$$\bar{M}_0 \leq \underline{\mathrm{M}}_l \leq L \overset{(22)}{\leq} \max\{\underline{\mathrm{M}}_l, \kappa\bar{L}\} \overset{(26)}{\leq} \max\left\{\bar{M}_0, \kappa\bar{L}\right\}$$

which implies that L satisfies the second relation in (19). To see the first relation in (19), observe that the first conclusion of Lemma 2.4 and the facts that $\xi = \xi_j$ and $y = y_j$ imply that $\phi(\xi) \leq \phi(y)$. It also follows from combining relations (25) and (28) that $\phi(\xi) \leq \phi(z_0)$, which together with the above relation implies that the first relation in (19) holds. Proposition 2.1(b) then immediately follows from the above conclusions. □

A.2 Proof of Proposition 2.1(c)

For the remainder of this subsection, consider the l-th cycle of RPF-SFISTA and assume that $j \geq 1$ is an iteration index generated during the cycle and that $\mu = \mu_{l-1}$ is in the interval $(0, \bar{\mu}]$.

The main novelty in the proof of Proposition 2.1(c) lies in our careful construction of a sequence γ_j which lower bounds the entire composite function ϕ. This construction is thus essential to establishing that RPF-SFISTA is $\bar{\mu}$-universal as opposed to just $\bar{\mu}_f$-universal.

The proofs of Lemmas A.2 and A.3 below follow closely to the proofs of Lemmas A.6 and A.7 in [38] except for minor modifications due to our novel construction of the sequence γ_j. We include the proofs here for completeness. The first lemma establishes key properties of the iterates of RPF-SFISTA.

Lemma A.2. *For every $j \geq 1$ and $x \in \mathbb{E}$, define*

$$\gamma_j(x) := \phi(y_j) + 2\left[\ell_f(y_j, \tilde{x}_{j-1}) - f(y_j)\right] + \langle s_j, x - y_j\rangle + \frac{\mu}{4}\|x - y_j\|^2, \tag{52}$$

where $\phi := f + h$ and s_j are as in (1) and (11), respectively. Then, for every $j \geq 1$, we have:

$$y_j = \operatorname*{argmin}_x \left\{\gamma_j(x) + \frac{L_j}{2}\|x - \tilde{x}_{j-1}\|^2\right\}; \tag{53}$$

$$x_j = \operatorname*{argmin}_x \left\{a_{j-1}\gamma_j(x) + \tau_{j-1}\|x - x_{j-1}\|^2/2\right\}. \tag{54}$$

Proof. It follows from the definition of s_j in (11) and the fact that $\nabla\gamma_j(y_j) = s_j$ that y_j satisfies the optimality condition for (53). This observation implies that relation (53) must hold. Moreover, it follows from relations (10) and (12) that

$$\begin{aligned} a_{j-1}\nabla\gamma_j(x_j) + \tau_{j-1}(x_j - x_{j-1}) &= a_{j-1}s_j + \frac{a_{j-1}\mu}{2}(x_j - y_j) + \tau_{j-1}(x_j - x_{j-1}) \\ &\overset{(10)}{=} a_{j-1}s_j - \frac{\mu a_{j-1}}{2}y_j - \tau_{j-1}x_{j-1} + \tau_j x_j \overset{(12)}{=} 0 \end{aligned}$$

which implies that relation (54) holds. □

Lemma A.3. *For every $j \geq 1$ and $x \in \mathbb{E}$, we have*

$$\begin{aligned} &A_{j-1}\gamma_j(y_{j-1}) + a_{j-1}\gamma_j(x) + \frac{\tau_{j-1}}{2}\|x_{j-1} - x\|^2 - \frac{\tau_j}{2}\|x_j - x\|^2 \\ &\qquad \geq A_j\phi(y_j) + \frac{\chi A_j L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2. \end{aligned} \tag{55}$$

Proof. Relation (54), the second relation in (10), the fact that $\Psi_j := a_{j-1}\gamma_j(\cdot) + \tau_{j-1}\|\cdot - x_{j-1}\|^2/2$ is $(\tau_{j-1} + \mu a_{j-1}/2)$-convex, and relation (41) with $\Psi = \Psi_j$ and $\nu = \tau_j$ imply that

$$a_{j-1}\gamma_j(x) + \frac{\tau_{j-1}}{2}\|x - x_{j-1}\|^2 - \frac{\tau_j}{2}\|x - x_j\|^2 \geq a_{j-1}\gamma_j(x_j) + \frac{\tau_{j-1}}{2}\|x_j - x_{j-1}\|^2 \quad \forall x \in \mathbb{E}. \tag{56}$$

It then follows from the convexity of γ_j, the definitions of $\tilde{x}_{j-1}$ and A_j in (6) and (10), respectively, and the second equality in (21), that

$$\begin{aligned} &A_{j-1}\gamma_j(y_{j-1}) + a_{j-1}\gamma_j(x_j) + \frac{\tau_{j-1}}{2}\|x_j - x_{j-1}\|^2 \\ &\quad \geq A_j\gamma_j\left(\frac{A_{j-1}y_{j-1} + a_{j-1}x_j}{A_j}\right) + \frac{\tau_{j-1}A_j^2}{2a_{j-1}^2}\left\|\frac{A_{j-1}y_{j-1} + a_{j-1}x_j}{A_j} - \frac{A_{j-1}y_{j-1} + a_{j-1}x_{j-1}}{A_j}\right\|^2 \\ &\quad \overset{(6)}{\geq} A_j \min_x\left[\gamma_j(x) + \frac{\tau_{j-1}A_j}{2a_{j-1}^2}\|x - \tilde{x}_{j-1}\|^2\right] \\ &\quad \overset{(21)}{=} A_j \min_x\left\{\gamma_j(x) + \frac{L_j}{2}\|x - \tilde{x}_{j-1}\|^2\right\} \end{aligned}$$

$$\overset{(53)}{=} A_j \left[\gamma_j(y_j) + \frac{L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2\right]$$
$$\overset{(52)}{=} A_j \left[\phi(y_j) + 2[\ell_f(y_j, \tilde{x}_{j-1}) - f(y_j)] + \frac{L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2\right]$$
$$\overset{(8)}{\geq} A_j \left[\phi(y_j) + \frac{\chi L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2\right]. \tag{57}$$

Combining relations (56) and (57) then immediately implies the conclusion of the lemma. □

We now present a technical lemma that is important for proving the lemma that directly follows it. The proof can be found in [20].

Lemma A.4. *Assume that φ is a ξ-strongly convex function and let $(y, \eta) \in \mathbb{E} \times \Re$ be such that $0 \in \partial_\eta \varphi(y)$. Then,*

$$0 \in \partial_{2\eta}\left(\varphi(\cdot) - \frac{\xi}{4}\|\cdot - y\|^2\right)(y).$$

The following lemma establishes that the estimate sequence γ_j constructed in (A.2) lower bounds ϕ.

Lemma A.5. *For every $j \geq 1$, we have that $\gamma_j \leq \phi$.*

Proof. For every $j \geq 1$, define

$$\tilde{\gamma}_j(x) := \ell_f(x; \tilde{x}_{j-1}) + h(x). \tag{58}$$

The fact that f is convex implies that $\tilde{\gamma}_j \leq \phi$. It follows from the definition of y_j in (7) that

$$y_j = \operatorname*{argmin}_x \left\{\tilde{\gamma}_j(x) + \frac{L_j}{2}\|x - \tilde{x}_{j-1}\|^2\right\}. \tag{59}$$

Now, it is easy to see from definition of s_j in (11) and relation (59) that $s_j \in \partial\tilde{\gamma}_j(y_j)$. Thus, by the subgradient inequality and the fact that $\tilde{\gamma}_j(x) \leq \phi(x)$, we have that for all $x \in \mathbb{E}$:

$$\phi(x) \geq \tilde{\gamma}_j(x) \geq \tilde{\gamma}_j(y_j) + \langle s_j, x - y_j\rangle = \phi(y_j) + \langle s_j, x - y_j\rangle - \eta_j,$$

where $\eta_j := \phi(y_j) - \tilde{\gamma}(y_j)$. Hence, by the definition of η_j-subdifferential, it follows that $s_j \in \partial_{\eta_j}\phi(y_j)$. Now, note that ϕ is a μ-strongly convex function since $\mu = \mu_{l-1}$ is assumed to be in the interval $(0, \bar{\mu}]$. Thus, by Lemma A.4 with $\xi = \mu$ and $\varphi(\cdot) = \phi(\cdot) - \langle s_j, \cdot - y_j\rangle$, we have

$$s_j \in \partial_{2\eta_j}\left(\phi(\cdot) - \frac{\mu}{4}\|\cdot - y_j\|^2\right)(y_j). \tag{60}$$

Hence, it follows from the definition of η_j above, relation (60), the fact that $h = \phi - f$, and the definitions of $\tilde{\gamma}_j(\cdot)$ and $\gamma_j(\cdot)$ in (58) and (52), respectively that

$$\phi(x) \overset{(60)}{\geq} \phi(y_j) + \langle s_j, x - y_j\rangle + \frac{\mu}{4}\|x - y_j\|^2 - 2\eta_j$$
$$= 2\tilde{\gamma}_j(y_j) - \phi(y_j) + \langle s_j, x - y_j\rangle + \frac{\mu}{4}\|x - y_j\|^2$$
$$\overset{(58)}{=} 2\ell_f(y_j; \tilde{x}_{j-1}) + 2h(y_j) - \phi(y_j) + \langle s_j, x - y_j\rangle + \frac{\mu}{4}\|x - y_j\|^2$$
$$= 2\ell_f(y_j; \tilde{x}_{j-1}) + 2[\phi(y_j) - f(y_j)] - \phi(y_j) + \langle s_j, x - y_j\rangle + \frac{\mu}{4}\|x - y_j\|^2 \overset{(52)}{=} \gamma_j(x),$$

from which statement of the lemma immediately follows. □

Lemma A.6. *For every $j \geq 1$ and $x \in \mathcal{H}$, we have*

$$\sigma_{j-1}(x) - \sigma_j(x) \geq \frac{\chi A_j L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2$$

where

$$\sigma_j(x) := A_j[\phi(y_j) - \phi(x)] + \frac{\tau_j}{2}\|x - x_j\|^2.$$

Proof. Using Lemma A.3 and Lemma A.5 we have

$$\begin{aligned} A_{j-1}\phi(y_{j-1}) + a_{j-1}\phi(x) + \frac{\tau_{j-1}}{2}\|x_{j-1} - x\|^2 - \frac{\tau_j}{2}\|x_j - x\|^2 \\ \geq A_j\phi(y_j) + \frac{\chi A_j L_j}{2}\|y_j - \tilde{x}_{j-1}\|^2. \end{aligned}$$

The conclusion of the lemma now follows by subtracting $A_j\phi(x)$ from both sides of the above inequality, and using the first equality in (10) and the definition of $\sigma_j(x)$. □

The following result is important for proving Proposition 2.1(c)

Lemma A.7. *For every $j \geq 2$ and $x \in \mathcal{H}$, it holds that*

$$A_{j-1}[\phi(\xi_{j-1}) - \phi(x)] + \frac{\tau_{j-1}}{2}\|x - x_{j-1}\|^2 \leq \frac{1}{2}\|x - x_0\|^2 - \frac{\chi}{2}\sum_{i=1}^{j-1} A_i L_i \|y_i - \tilde{x}_{i-1}\|^2. \tag{61}$$

Proof. It follows from summing the inequality of Lemma A.6 from $j = 1$ to $j = j - 1$, using the facts that $A_0 = 0$ and $\tau_0 = 1$, and using the definition of $\sigma_j(\cdot)$ in Lemma A.6 that

$$A_{j-1}[\phi(y_{j-1}) - \phi(x)] + \frac{\tau_{j-1}}{2}\|x - x_{j-1}\|^2 \leq \frac{1}{2}\|x - x_0\|^2 - \frac{\chi}{2}\sum_{i=1}^{j-1} A_i L_i \|y_i - \tilde{x}_{i-1}\|^2.$$

Relation (61) then immediately follows from the above relation and the fact that the first conclusion of Lemma 2.4 implies that $\phi(\xi_{j-1}) \leq \phi(y_{j-1})$. □

We are now ready to prove Proposition 2.1(c).

Proof of Proposition 2.1(c). Consider the l-th cycle of RPF-SFISTA and assume that it is performed with $\mu_{l-1} \in (0, \bar{\mu}]$. Using relation (61) with $x = \xi_{j-1}$, it follows that

$$\|\xi_{j-1} - x_0\|^2 \overset{(61)}{\geq} \chi \sum_{i=1}^{j-1} A_i L_i \|y_i - \tilde{x}_{i-1}\|^2 \geq \chi A_{j-1} L_{j-1} \|y_{j-1} - \tilde{x}_{j-2}\|^2.$$

It then follows from the above relation that for any iteration index $j \geq 1$ generated during the l-th cycle, it holds that

$$\|\xi_j - x_0\|^2 \geq \chi A_j L_j \|y_j - \tilde{x}_{j-1}\|^2. \tag{62}$$

Hence, relation (62) implies that the inequality (14) checked in step 4 of RPF-SFISTA always holds for every iteration index $j \geq 1$ generated during the l-th cycle and hence the l-th cycle of RPF-SFISTA never terminates in step 4. This observation together with Proposition 2.1(a)-(b) then immediately imply that the l-th cycle must terminate successfully in its step 5 with a quadruple (y, v, ξ, L) that satisfies (19) and such that (y, v) is an $\hat{\epsilon}$-optimal solution of (1) in at most (18) ACG iterations/resolvent evaluations. □

References

[1] Teodoro Alamo, Pablo Krupa, and Daniel Limon. Gradient based restart fista. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3936–3941, 2019.

[2] Teodoro Alamo, Pablo Krupa, and Daniel Limon. Restart of accelerated first-order methods with linear convergence under a quadratic functional growth condition. *IEEE Transactions on Automatic Control*, 68(1):612–619, 2023.

[3] Teodoro Alamo, Daniel Limon, and Pablo Krupa. Restart fista with global linear convergence. In *2019 18th European Control Conference (ECC)*, pages 1969–1974, 2019.

[4] J. Aujol, C. Dossal, H. Labarrière, and A. Rondepierre. Fista restart using an automatic estimation of the growth parameter. *hal-03153525v4*, 2022.

[5] Jean-Franćois Aujol, Luca Calatroni, Charles Dossal, Hippolyte Labarrière, and Aude Rondepierre. Parameter-free fista by adaptive restart and backtracking. *SIAM Journal on Optimization*, 34(4):3259–3285, 2024.

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

[7] L. Calatroni and A. Chambolle. Backtracking strategies for accelerated descent methods with smooth composite objectives. *SIAM J. Optim.*, 29(3):1772–1798, 2019.

[8] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM J. Optim.*, 28(2):1751–1772, 2018.

[9] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

[10] O. Fercoq and Z. Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *IMA Journal of Numerical Analysis*, 39:2069–2095, 2019.

[11] 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.

[12] 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(4):2182–2211, 2015.

[13] W. Kong, J.G. Melo, and R.D.C. Monteiro. Complexity of a quadratic penalty accelerated inexact proximal point method for solving linearly constrained nonconvex composite programs. *SIAM J. Optim.*, 29(4):2566–2593, 2019.

[14] Weiwei 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.

[15] G. Lan and R.D.C Monteiro. Iteration-complexity of first-order penalty methods for convex programming. *Technical Report. Optimization Online.*, 2008.

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

[17] G. Lan, Y. Ouyang, and Z. Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization. *Available on arXiv:2310.12139*, 2023.

[18] P. Latafat, A. Themelis, L. Stella, and P. Patrinos. Adaptive proximal algorithms for convex optimization under local lipschitz continuity of the gradient. *Available on arXiv:2301.04431*, 2023.

[19] T. Li and G. Lan. A simple uniformly optimal method without line search for convex optimization. *Available on arXiv:2310.10082*, 2023.

[20] J. Liang and R.D.C. Monteiro. A Doubly Accelerated Inexact Proximal Point Method for Nonconvex Composite Optimization Problems. *Available on arXiv:1811.11378v2*, 2018.

[21] J. Liang and R.D.C. Monteiro. Average curvature fista for nonconvex smooth composite optimization problems. *Comput. Optim. Appl.*, 86:275–302, 2023.

[22] Jiaming Liang and Renato D. C. Monteiro. An average curvature accelerated composite gradient method for nonconvex smooth composite optimization problems. *SIAM Journal on Optimization*, 31(1):217–243, 2021.

[23] Jingwei Liang, Tao Luo, and Carola-Bibiane Schönlieb. Improving "fast iterative shrinkage-thresholding algorithm": Faster, smarter, and greedier. *SIAM Journal on Scientific Computing*, 44(3):A1069–A1091, 2022.

[24] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[25] Q. Lin and L. Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Comput. Optim. Appl.*, 60:633–674, 2015.

[26] Y. Malitsky and K. Mishchenko. Adaptive proximal gradient method for convex optimization. *Available on arXiv:2308.02261*, 2023.

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

[28] W. Moursi, V. Pavlovic, and S. Vavasis. Accelerated gradient descent: A guaranteed bound for a heuristic restart strategy. *Available on arXiv:2310.07674*, 2023.

[29] I. Necoara, Y. Nesterov, and F. Gilneur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical programming*, 175(1):69–107, 2019.

[30] Y. Nesterov. How to make the gradients small. *OPTIMA, MPS Newsletter*, (88):10–11, 2012.

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

[32] Y.E. Nesterov. Gradient methods for minimizing composite functions. *Math. Program.*, pages 1–37, 2012.

[33] B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15:715–732, 2015.

[34] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst for gradient-based nonconvex optimization. In *AISTATS 2018-21st International Conference on Artificial Intelligence and Statistics*, pages 1–10, 2018.

[35] J. Renegar and B. Grimmer. A simple nearly optimal restart scheme for speeding up first-order methods. *Foundations of computational mathematics*, 22(1):211–256, 2022.

[36] Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020.

[37] K. Scheinberg, D. Goldfarb, and X. Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of computational mathematics*, 14:389–417, 2014.

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