A FISTA-type accelerated gradient algorithm for solving smooth nonconvex composite optimization problems

Jiaming Liang * Renato D.C. Monteiro * Chee-Khian Sim †

Abstract

In this paper, we describe and establish iteration-complexity of two accelerated composite gradient (ACG) variants to solve a smooth nonconvex composite optimization problem whose objective function is the sum of a nonconvex differentiable function f with a Lipschitz continuous gradient and a simple nonsmooth closed convex function h. When f is convex, the first ACG variant reduces to the well-known FISTA for a specific choice of the input, and hence the first one can be viewed as a natural extension of the latter one to the nonconvex setting. The first variant requires an input pair (M, m) such that f is m-weakly convex, ∇f is M-Lipschitz continuous, and $m \leq M$ (possibly $m < M$), which is usually hard to obtain or poorly estimated. The second variant on the other hand can start from an arbitrary input pair (M, m) of positive scalars and its complexity is shown to be not worse, and better in some cases, than that of the first variant for a large range of the input pairs. Finally, numerical results are provided to illustrate the efficiency of the two ACG variants.

1 Introduction

Accelerated gradient methods for solving convex noncomposite programs were originally developed by Nesterov in his celebrated work [21]. Subsequently, several variants of this method (see for example [1, 15, 20, 22, 23, 27]) were developed for solving convex simple-constrained or composite programs, which we refer generically to as ACG variants. These variants have also been used as subroutines in several inexact-type proximal algorithms for solving convex-concave saddle point and monotone Nash equilibrium problems (see for example [4, 10, 11, 13, 23, 24]).

In this paper, we study ACG algorithms to solve the smooth nonconvex composite optimization (SNCO) problem

$$\phi_* := \min \{\phi(z) := f(z) + h(z) : z \in \mathbb{R}^n\} \tag{1}$$

where $h : \mathbb{R}^n \to (-\infty, \infty]$ is a proper lower-semicontinuous convex function with bounded $\operatorname{dom} h$ and f is a real-valued differentiable (possibly nonconvex) function whose gradient is M-Lipschitz continuous on $\operatorname{dom} h$, i.e., for every $z, z' \in \operatorname{dom} h$,

$$\|\nabla f(z') - \nabla f(z)\| \leq M\|z' - z\|. \tag{2}$$

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (email: jiaming.liang@gatech.edu and renato.monteiro@isye.gatech.edu). This work was partially supported by ONR Grant N00014-18-1-2077.

†School of Mathematics and Physics, University of Portsmouth, Lion Gate Building, Lion Terrace, Portsmouth PO1 3HF. (email: chee-khian.sim@port.ac.uk). This work is made possible through an LMS Research in Pairs (Scheme 4) grant.

The first analysis of an ACG algorithm for solving (1) under the above assumption appears in [6] where essentially a well-known ACG variant that solves the convex version of (1) is also shown to solve its nonconvex version in the following sense: for a given tolerance $\hat{\rho} > 0$, it computes $(\hat{y}, \hat{v}) \in \operatorname{dom} h \times \mathbb{R}^n$ such that $\hat{v} \in \nabla f(\hat{y}) + \partial h(\hat{y})$ and $\|\hat{v}\| \leq \hat{\rho}$ in

$$\mathcal{O}\left(\frac{M\bar{m}D_h^2}{\hat{\rho}^2} + \left(\frac{Md_0}{\hat{\rho}}\right)^{2/3}\right) \tag{3}$$

iterations where d_0 is the distance of the initial point x_0 to the optimal solution set of (1), D_h is the diameter of $\operatorname{dom} h$ and $\bar{m}$ is the smallest scalar $m \geq 0$ such that

$$-\frac{m}{2}\|z' - z\|^2 \leq f(z') - f(z) - \langle \nabla f(z), z' - z\rangle. \tag{4}$$

for every $z, z' \in \operatorname{dom} h$. Any pair (M, m) with $m \leq M$ and satisfying both (2) and (4) is referred to as a curvature pair. We refer to the ACG variant of [6] as the AG method and note that each one of its iterations performs exactly two resolvent evaluations of h, i.e., an evaluation of the point-to-point operator $(I + \tau\partial h)^{-1}(\cdot)$ for some $\tau > 0$. (Several examples of convex, as well as nonconvex, functions h whose resolvent evaluations are easy to compute can be found in [8].)

This paper describes and establishes the iteration-complexities of two ACG variants for solving the nonconvex version of (1). The first variant can be viewed as a direct extension of the FISTA presented in [1] for solving the convex version of (1). In contrast to an iteration of the AG method, every iteration of the first variant performs exactly one resolvent evaluation of h. One drawback of the first variant is that it requires as input a curvature pair (M, m), which is usually hard to obtain or is poorly estimated. Letting $(\bar{M}, \bar{m})$ denote the smallest curvature pair, a second variant is proposed to remedy the aforementioned drawback in that it works regardless of the choice of input pair (M, m) (i.e., not necessarily satisfying (2) and (4)), and its complexity is shown to be not worse than (3) when $M \geq \bar{M}$ and $m \in [\bar{m}, M]$. Moreover, when $m \in [\bar{m}, \bar{M}]$, the complexity of the second variant is empirically argued to behave as (3) with $M = \bar{M}$, for a large range of scalars M such that $M \leq \bar{M}$ (see the second paragraph following Theorem 3.4) and our computational results demonstrate that taking M relatively smaller than $\bar{M}$ can substantially improve its performance. It is also shown that all iterations of the second variant, with the exception of a few ones whose total number is log-bounded, perform exactly one resolvent evaluation of h.

Related works. Inspired by [6], other papers have proposed ACG variants for solving (1) under the assumption that f is a nonconvex continuously differentiable function with a Lipschitz continuous gradient, and that h is a simple lower semi-continuous convex (see e.g. [5, 7]) or nonconvex (see e.g. [16, 17, 29]) function. Similar to an iteration of the two ACG variants in our paper, the one of the algorithms in [17, 29] requires exactly one resolvent evaluation of h. However, while every iteration of the variants studied here is always accelerated, the ones of the latter algorithms can be a simple composite gradient (and unaccelerated) step whenever a certain descent property is not satisfied.

Another approach for solving (1) consists of using a descent unaccelerated inexact proximal-type method where each prox subproblem is constructed to be (possibly strongly) convex and hence solved by an ACG variant (see [3, 14, 25]). Moreover, the approach has the benefit of working with a larger prox stepsize and hence of having a better outer iteration-complexity than the approaches in the previous paragraph. However, each of its outer iterations still has to perform a uniformly bounded number of inner iterations to approximately solve a prox subproblem. Overall, it is shown that its inner-iteration complexity is better than the iteration-complexities of the methods in the

previous paragraph, particularly when $\bar{m} \ll \bar{M}$. As in the papers [5, 7, 16, 17, 29] in the previous paragraph, it is worth noting that the method in [25] attempts to perform an accelerated step whenever a certain descent property holds and, in case of failure, it performs an unaccelerated prox step similar to the one used in the methods in [3, 14].

Finally, a hybrid approach that borrows ideas from the above group of papers is presented in [18]. More specifically, the latter work presents an accelerated inexact proximal point method reminiscent of those presented in [9, 20, 26], but in which only the convex version of (1) is considered. Each (outer) iteration of the method requires that a prox subproblem be approximately solved by using an ACG variant in the same way as in the papers [3, 14]. Hence, similar to the methods in the previous paragraph, this method performs both outer and inner iterations with a major difference that every outer iteration is an accelerated step (as in the papers [5, 7, 16, 17, 29]) with a large proximal stepsize (as in the papers [3, 14]).

Organization of the paper. Subsection 1.1 presents basic definitions and notations used throughout the paper. Section 2 presents assumptions made on the SNCO problem, describes the first ACG variant, which is an extension of FISTA to the SNCO problem and is referred to as NC-FISTA, and establishes its iteration-complexity for obtaining a stationary point of the SNCO problem. Section 3 presents an adaptive variant of NC-FISTA, namely, ADAP-NC-FISTA, and establishes its iteration-complexity. Section 4 presents computational results showing the efficiency of NC-FISTA and ADAP-NC-FISTA. Section 5 finishes the paper by presenting a few concluding remarks. Finally, supplementary technical results are provided in the appendix.

1.1 Basic definitions and notation

This subsection provides some basic definitions and notations used in this paper.

The set of real numbers is denoted by $\mathbb{R}$. The set of non-negative real numbers and the set of positive real numbers are denoted by $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively. Let $\mathbb{R}^n$ denote the standard n-dimensional Euclidean space with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively. The Frobenius inner product and Frobenius norm in $\mathbb{R}^{m\times n}$ are denoted by $\langle \cdot, \cdot \rangle_F$ and $\|\cdot\|_F$, respectively. The sets of real $n \times n$ symmetric positive semidefinite matrices are denoted by S^n_+. Let $N_X(z)$ denote the normal cone of X at z, i.e., $N_X(z) = \{u \in \mathbb{R}^n : \langle u, z' - z\rangle \leq 0 \quad \forall z' \in X\}$. The indicator function I_X of a set $X \subset \mathbb{R}^n$ is defined as $I_X(z) = 0$ for every $z \in X$, and $I_X(z) = \infty$, otherwise. If Ω is a nonempty closed convex set, the orthogonal projection $P_\Omega : \mathbb{R}^n \to \mathbb{R}^n$ onto Ω is defined as

$$P_\Omega(z) := \operatorname{argmin}_{z' \in \Omega} \|z' - z\| \quad \forall z \in \mathbb{R}^n.$$

Define $\log^+(s) := \max\{\log s, 0\}$ and $\log_1^+(s) := \max\{\log s, 1\}$ for $s > 0$.

Let $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ be given. The effective domain of Ψ is denoted by $\operatorname{dom}\Psi := \{x \in \mathbb{R}^n : \psi(x) < \infty\}$ and Ψ is proper if $\operatorname{dom}\Psi \neq \emptyset$. Moreover, a proper function $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ is μ-strongly convex for some $\mu \geq 0$ if

$$\Psi(\beta z + (1-\beta)z') \leq \beta\Psi(z) + (1-\beta)\Psi(z') - \frac{\beta(1-\beta)\mu}{2}\|z - z'\|^2$$

for every $z, z' \in \operatorname{dom}\Psi$ and $\beta \in [0, 1]$. Let $\partial\Psi(z)$ denote the subdifferential of Ψ at $z \in \operatorname{dom}\Psi$. If Ψ is differentiable at $\bar{z} \in \mathbb{R}^n$, then its affine approximation $\ell_\Psi(\cdot; \bar{z})$ at $\bar{z}$ is defined as

$$\ell_\Psi(z; \bar{z}) := \Psi(\bar{z}) + \langle \nabla\Psi(\bar{z}), z - \bar{z}\rangle \quad \forall z \in \mathbb{R}^n.$$

Let $\overline{\operatorname{Conv}}(\mathbb{R}^n)$ denote the set of all proper lower semi-continuous convex functions $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$.

2 NC-FISTA for solving the SNCO problem

This section describes the assumptions made on our problem of interest, namely, problem (1). It also presents and establishes the iteration-complexity of the first ACG variant, namely NC-FISTA, for obtaining an approximate solution of (1).

Throughout this paper, we consider problem (1) and make the following assumptions on it:

(A1) $h \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$;

(A2) $\mathrm{dom}\, h$ is bounded;

(A3) f is differentiable on a closed convex set $\Omega \supseteq \mathrm{dom}\, h$ and there exists $M > 0$ such that (2) holds for every $z, z' \in \Omega$;

(A4) f is nonconvex on $\mathrm{dom}\, h$ and there exists $m > 0$ such that (4) holds for every $z, z' \in \Omega$.

Throughout this paper, we denote the diameter of $\mathrm{dom}\, h$ as

$$D_h := \sup\{\|u' - u\| : u, u' \in \mathrm{dom}\, h\} < \infty \tag{5}$$

where its finiteness is due to (A2). Moreover, let $\bar{M}$ (resp., $\bar{m}$) denote the smallest scalar M (resp., m) satisfying (2) (resp., (4)) for every $z, z' \in \Omega$. Clearly, $\bar{M} \geq \bar{m} > 0$.

We now make a few remarks about the above assumptions. First, (A1)-(A3) imply that the set Z^* of optimal solutions of (1) is nonempty and compact. Second, using the fact that $\bar{M}$ satisfies (2) for every $z, z' \in \Omega$ in view of the above definition of $\bar{M}$, we easily see that

$$\big|f(z') - \ell_f(z'; z)\big| \leq \frac{\bar{M}}{2}\|z' - z\|^2 \quad \forall z, z' \in \Omega,$$

and hence that (4) is satisfied with $m = \bar{M}$. Thus, it follows that from the definition of $\bar{m}$ that $\bar{m} \leq \bar{M}$. Third, (A4) implies that $\bar{m} > 0$. Fourth, our interest is in the case where $\bar{m} \ll \bar{M}$ since this case naturally arises in the context of penalty methods for solving linearly constrained composite nonconvex optimization problems (e.g., see Section 4 of [14]).

For $z \in \mathrm{dom}\, h$ to be a local minimizer of (1), a necessary condition is that z is a stationary point of (1), i.e., $0 \in \nabla f(z) + \partial h(z)$. Motivated by this remark, the following notion of an approximate solution to problem (1) is proposed: a pair $(\hat{y}, \hat{v})$ is said to be a $\hat{\rho}$-approximate solution to (1), for a given tolerance $\hat{\rho} > 0$, if

$$\hat{v} \in \nabla f(\hat{y}) + \partial h(\hat{y}), \quad \|\hat{v}\| \leq \hat{\rho}. \tag{6}$$

We are now ready to state the NC-FISTA for solving (1).

NC-FISTA

---

0. Let an initial point $y_0 \in \mathrm{dom}\, h$, a pair $(M, m) \in \mathbb{R}^2_{++}$ such that $M \geq m \geq \bar{m}$ and $M > \bar{M}$, a scalar $A_0 > 0$, and a tolerance $\hat{\rho} > 0$ be given, and set $x_0 = y_0$, $\lambda = 1/M$, $k = 0$ and

$$\kappa_0 = \frac{1 + \sqrt{1 + 4A_0}}{\sqrt{1 + 4A_0} - 1}; \tag{7}$$

1. compute

$$a_k = \frac{1 + \sqrt{1 + 4A_k}}{2}, \quad A_{k+1} = A_k + a_k; \tag{8}$$

2. compute

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_k}{A_{k+1}} x_k \tag{9}$$

$$y_{k+1} = \text{argmin}_u \left\{ \ell_f(u; \tilde{x}_k) + h(u) + \frac{1}{2} \left(\frac{1}{\lambda} + \frac{\kappa_0 m}{a_k} \right) \|u - \tilde{x}_k\|^2 \right\}, \tag{10}$$

$$\hat{x}_{k+1} = \frac{(a_k + \kappa_0 m\lambda) y_{k+1} - (a_k - 1) y_k}{\kappa_0 m\lambda + 1}, \quad x_{k+1} = P_\Omega\left(\hat{x}_{k+1}\right); \tag{11}$$

3. compute

$$v_{k+1} = \left(\frac{1}{\lambda} + \frac{\kappa_0 m}{a_k} \right) (\tilde{x}_k - y_{k+1}) + \nabla f(y_{k+1}) - \nabla f(\tilde{x}_k); \tag{12}$$

if $\|v_{k+1}\| \leq \hat{\rho}$ then output $(\hat{y}, \hat{v}) = (y_{k+1}, v_{k+1})$ and **stop**; otherwise, set $k \leftarrow k+1$ and go to step 1.

We now make a few remarks about the NC-FISTA. First, it follows from (10) that $\{y_k\} \subset \text{dom}\, h$, and hence $\{y_k\}$ is bounded in view of (A2). Second, the definition of $\{x_k\}$ in (11) implies that $\{x_k\} \subset \Omega$, and hence that $\{\tilde{x}_k\} \subset \Omega$ in view of (9). Hence, if Ω is chosen to be compact, then the latter two sequences will also be bounded but our analysis does not make such an assumption on Ω. Third, if $\Omega = \mathbb{R}^n$, then each iteration of the NC-FISTA requires one resolvent evaluation of h in (10), i.e., an evaluation of $(I + \tau\partial h)^{-1}$ for some $\tau > 0$. Otherwise, it requires an extra projection onto Ω in (11) , which, depending on the problem instance and the set Ω, might be considerably cheaper than a resolvent evaluation of h. Fourth, it follows from (8) that $\{a_k\}$ and $\{A_k\}$ are strictly increasing sequences of positive scalars. Fifth, A_0 is required to be positive so as to guarantee that the quantity κ_0 defined in (7) is well-defined. We will assume later on that $A_0 = \Theta(1)$ so as to eliminate it from the iteration-complexity bounds for NC-FISTA. Sixth, NC-FISTA requires that M and m be upper bounds for $\bar{M}$ and $\bar{m}$, respectively, due to technical requirements that appear in its iteration-complexity analysis. Actually, M is also required to be not too close to $\bar{M}$. Seventh, if a scalar M is known, then setting m to be equal to M fulfills the conditions of step 0 of NC-FISTA in view of the fact that $\bar{M} \geq \bar{m}$. However, NC-FISTA also allows for the possibility that a sharper scalar $m \in [\bar{m}, M)$ is known due to the fact that its iteration-complexity bound improves as m decreases (see Theorem 2.6). Eighth, when f is convex, i.e., $\bar{m} = 0$, NC-FISTA reduces to FISTA if m is set to zero. Finally, (8) implies that

$$A_{k+1} = a_k^2. \tag{13}$$

We establish a number of technical results. The first one establishes an important inequality satisfied by m.

Lemma 2.1 *For $k \geq 0$, we have*

$$\frac{\bar{m}}{\kappa_0} + \frac{m}{a_k} \leq m.$$

Proof: Using the assumption $m \geq \bar{m}$, the definition of κ_0 in (7), relation (8) with $k = 0$, and the fact that $\{a_k\}$ is increasing, we conclude that for every $k \geq 0$,

$$m - \frac{\bar{m}}{\kappa_0} \geq \left(1 - \frac{1}{\kappa_0}\right) m = \frac{2m}{1 + \sqrt{1 + 4A_0}} = \frac{m}{a_0} \geq \frac{m}{a_k}.$$

■

The following results introduce two functions that play important roles in our analysis of NC-FISTA and establish some basic facts about them.

Lemma 2.2 *For every $k \geq 0$, if we define*

$$\tilde{\gamma}_k(u) := \ell_f(u; \tilde{x}_k) + h(u) + \frac{\kappa_0 m}{2a_k}\|u - \tilde{x}_k\|^2, \tag{14}$$

$$\gamma_k(u) := \tilde{\gamma}_k(y_{k+1}) + \frac{1}{\lambda}\langle \tilde{x}_k - y_{k+1}, u - y_{k+1}\rangle + \frac{\kappa_0 m}{2a_k}\|u - y_{k+1}\|^2, \tag{15}$$

then the following statements hold:

(a) both γ_k and $\tilde{\gamma}_k$ are $(\kappa_0 m/a_k)$-strongly convex functions, γ_k minorizes $\tilde{\gamma}_k$, $\tilde{\gamma}_k(y_{k+1}) = \gamma_k(y_{k+1})$,

$$\min_u \left\{\tilde{\gamma}_k(u) + \frac{1}{2\lambda}\|u - \tilde{x}_k\|^2\right\} \;=\; \min_u \left\{\gamma_k(u) + \frac{1}{2\lambda}\|u - \tilde{x}_k\|^2\right\}, \tag{16}$$

and these minimization problems have y_{k+1} as a unique optimal solution;

(b) for every $u \in \operatorname{dom} h$,

$$\tilde{\gamma}_k(u) - \phi(u) \leq \frac{1}{2}\left(\bar{m} + \frac{\kappa_0 m}{a_k}\right)\|u - \tilde{x}_k\|^2;$$

(c) $x_{k+1} = \operatorname{argmin}_{u\in\Omega}\left\{a_k\gamma_k(u) + \|u - x_k\|^2/(2\lambda)\right\}$.

Proof: (a) It clearly follows from (15) that $\gamma_k(y_{k+1}) = \tilde{\gamma}_k(y_{k+1})$. By definitions of $\tilde{\gamma}_k$ and γ_k in (14) and (15) respectively, they are clearly $(\kappa_0 m/a_k)$-strongly convex. By (10) and the definition of $\tilde{\gamma}_k$ in (14), y_{k+1} is the optimal solution to the first minimization problem in (16). Since the objective function of this minimization problem is $[(1/\lambda) + (\kappa_0 m/a_k)]$-strongly convex, it follows that for all $u \in \mathbb{R}^n$,

$$\tilde{\gamma}_k(y_{k+1}) + \frac{1}{2\lambda}\|y_{k+1} - \tilde{x}_k\|^2 + \frac{1}{2}\left(\frac{1}{\lambda} + \frac{\kappa_0 m}{a_k}\right)\|y_{k+1} - u\|^2 \leq \tilde{\gamma}_k(u) + \frac{1}{2\lambda}\|u - \tilde{x}_k\|^2. \tag{17}$$

On the other hand, the definition of γ_k in (15) and the relation

$$\|y_{k+1} - \tilde{x}_k\|^2 + \|y_{k+1} - u\|^2 - \|u - \tilde{x}_k\|^2 = 2\langle \tilde{x}_k - y_{k+1}, u - y_{k+1}\rangle.$$

imply that

$$\tilde{\gamma}_k(y_{k+1}) + \frac{1}{2\lambda}\|y_{k+1} - \tilde{x}_k\|^2 + \frac{1}{2}\left(\frac{1}{\lambda} + \frac{\kappa_0 m}{a_k}\right)\|y_{k+1} - u\|^2 = \gamma_k(u) + \frac{1}{2\lambda}\|u - \tilde{x}_k\|^2. \tag{18}$$

Thus, it follows from (17) and (18) that $\gamma_k \leq \tilde{\gamma}_k$. Noting that the objective function in the second minimization problem in (16) is quadratic and using the first order optimality condition, we show that y_{k+1} is a unique optimal solution to the aforementioned problem.

(b) This statement follows from the assumption (A4) and the definition of $\tilde{\gamma}_k(u)$ in (14).

(c) Using the expressions for $\tilde{x}_k$ and $\hat{x}_{k+1}$ in (9) and (11), respectively, it is easy to see that $\hat{x}_{k+1}$ is the (unique) global minimizer of the function $a_k\gamma_k(u) + \|u - x_k\|^2/(2\lambda)$ over the whole space $\mathbb{R}^n$. The definition of x_{k+1} and the previous observation then imply that the conclusion of (c) holds. ■

The following result states a recursive inequality that plays an important role in the convergence rate analysis of NC-FISTA.

Lemma 2.3 *For every $u \in \Omega$ and $k \geq 0$, we have*

$$\begin{aligned}&\lambda A_{k+1}\phi(y_{k+1}) + \frac{\kappa_0 m\lambda + 1}{2}\|u - x_{k+1}\|^2 + \frac{(1-\lambda\mathcal{C}_k)A_{k+1}}{2}\|y_{k+1} - \tilde{x}_k\|^2 \\ &\leq \lambda A_k\gamma_k(y_k) + \lambda a_k\gamma_k(u) + \frac{1}{2}\|u - x_k\|^2,\end{aligned}$$

where

$$\mathcal{C}_k := \frac{2\left[f(y_{k+1}) - \ell_f(y_{k+1};\tilde{x}_k)\right]}{\|y_{k+1} - \tilde{x}_k\|^2}.$$

Proof: Using the definition of $\mathcal{C}_k$, (14) and Lemma 2.2(a), we conclude that

$$\begin{aligned}\lambda\phi(y_{k+1}) + \frac{1-\lambda\mathcal{C}_k}{2}\|y_{k+1} - \tilde{x}_k\|^2 &= \lambda\tilde{\gamma}_k(y_{k+1}) + \left(\frac{1}{2} - \frac{\kappa_0 m\lambda}{2a_k}\right)\|y_{k+1} - \tilde{x}_k\|^2 \\ &\leq \lambda\tilde{\gamma}_k(y_{k+1}) + \frac{1}{2}\|y_{k+1} - \tilde{x}_k\|^2 = \lambda\gamma_k(y_{k+1}) + \frac{1}{2}\|y_{k+1} - \tilde{x}_k\|^2. \qquad (19)\end{aligned}$$

On the other hand, using the fact that γ_k is convex, y_{k+1} is an optimal solution of (16), and relations (9) and (13), we conclude that for every $u \in \Omega$,

$$\begin{aligned}&A_{k+1}\left(\lambda\gamma_k(y_{k+1}) + \frac{1}{2}\|y_{k+1} - \tilde{x}_k\|^2\right) \\ &\leq A_{k+1}\left(\lambda\gamma_k\left(\frac{A_k y_k + a_k x_{k+1}}{A_{k+1}}\right) + \frac{1}{2}\left\|\frac{A_k y_k + a_k x_{k+1}}{A_{k+1}} - \tilde{x}_k\right\|^2\right) \\ &\leq \lambda A_k\gamma_k(y_k) + \lambda a_k\gamma_k(x_{k+1}) + \frac{A_{k+1}}{2}\left\|\frac{A_k y_k + a_k x_{k+1}}{A_{k+1}} - \tilde{x}_k\right\|^2 \\ &= \lambda A_k\gamma_k(y_k) + \lambda a_k\gamma_k(x_{k+1}) + \frac{1}{2}\|x_{k+1} - x_k\|^2 \\ &\leq \lambda A_k\gamma_k(y_k) + \lambda a_k\gamma_k(u) + \frac{1}{2}\|u - x_k\|^2 - \frac{\kappa_0 m\lambda + 1}{2}\|u - x_{k+1}\|^2, \qquad (20)\end{aligned}$$

where the last inequality follows from Lemma 2.2(c), the fact that γ_k is $(\kappa_0 m/a_k)$-strongly convex in view of Lemma 2.2(a), and hence that $\lambda a_k\gamma_k(u) + \|u - x_k\|^2/2$ is $(\kappa_0 m\lambda + 1)$-strongly convex. The result now follows by combining (19) and (20). ∎

Lemma 2.4 *For every $k \geq 1$ and $u \in \operatorname{dom} h$, we have*

$$\begin{aligned}\sum_{i=0}^{k-1}(1-\lambda\mathcal{C}_i)A_{i+1}\|y_{i+1} - \tilde{x}_i\|^2 &\leq 2\lambda A_0(\phi(y_0) - \phi(u)) - 2\lambda A_k(\phi(y_k) - \phi(u)) \\ &\quad + (\kappa_0 m\lambda + 1)\left(\|u - x_0\|^2 - \|u - x_k\|^2\right) + \kappa_0 m\lambda D_h^2 k + \bar{m}\lambda D_h^2\sum_{i=0}^{k-1} a_i. \qquad (21)\end{aligned}$$

Proof: Let $i \geq 0$ and $u \in \operatorname{dom} h$ be given. It follows from Lemma 2.2(a)-(b) that we have

$$\gamma_i(u) - \phi(u) \leq \tilde{\gamma}_i(u) - \phi(u) \leq \frac{1}{2}\left(\bar{m} + \frac{\kappa_0 m}{a_i}\right)\|u - \tilde{x}_i\|^2. \qquad (22)$$

Note that for every $A, a \in \mathbb{R}_+$ and $x, y \in \mathbb{R}^n$, we have

$$A\|y\|^2 + a\|x\|^2 = (A+a)\left\|\frac{Ay+ax}{A+a}\right\|^2 + \frac{Aa}{A+a}\|y-x\|^2.$$

Applying the above identity with $A = A_i$, $a = a_i$, $y = y_i - \tilde{x}_i$ and $x = u - \tilde{x}_i$, and using the definition of $\tilde{x}_i$ in (9) and the relation (13), we obtain

$$\begin{aligned} A_i\|y_i - \tilde{x}_i\|^2 + a_i\|u - \tilde{x}_i\|^2 &= A_{i+1}\left\|\frac{A_i y_i + a_i u}{A_{i+1}} - \tilde{x}_i\right\|^2 + \frac{A_i a_i}{A_{i+1}}\|y_i - u\|^2 \\ &= \|u - x_i\|^2 + \frac{A_i a_i}{A_{i+1}}\|y_i - u\|^2 \le \|u - x_i\|^2 + a_i D_h^2. \end{aligned} \tag{23}$$

where the inequality follows from the fact that $A_{i+1} = A_i + a_i \ge A_i$ due to (8) and the definition of D_h in (5).

Now, using Lemma 2.3, relations (8), (22) and (23), and some simple algebraic manipulations, we conclude that for every $i \ge 0$,

$$\begin{aligned} (1-\lambda\mathcal{C}_i)A_{i+1}\|y_{i+1} - \tilde{x}_i\|^2 &+ (\kappa_0 m\lambda + 1)\|u - x_{i+1}\|^2 - \|u - x_i\|^2 \\ &+ 2\lambda A_{i+1}(\phi(y_{i+1}) - \phi(u)) - 2\lambda A_i(\phi(y_i) - \phi(u)) \\ &\le 2\lambda A_i(\gamma_i(y_i) - \phi(y_i)) + 2\lambda a_i(\gamma_i(u) - \phi(u)) \\ &\le \lambda\left(\bar{m} + \frac{\kappa_0 m}{a_i}\right)\left(A_i\|y_i - \tilde{x}_i\|^2 + a_i\|u - \tilde{x}_i\|^2\right) \\ &\le \lambda\left(\bar{m} + \frac{\kappa_0 m}{a_i}\right)\left(\|u - x_i\|^2 + a_i D_h^2\right) \\ &= \lambda\left(\bar{m} + \frac{\kappa_0 m}{a_i}\right)\|u - x_i\|^2 + (\bar{m}a_i + \kappa_0 m)\lambda D_h^2. \end{aligned}$$

It follows from the above inequality and Lemma 2.1 that

$$\begin{aligned} (1-\lambda\mathcal{C}_i)A_{i+1}\|y_{i+1} - \tilde{x}_i\|^2 &+ 2\lambda A_{i+1}(\phi(y_{i+1}) - \phi(u)) + (\kappa_0 m\lambda + 1)\|u - x_{i+1}\|^2 \\ &\le 2\lambda A_i(\phi(y_i) - \phi(u)) + (\kappa_0 m\lambda + 1)\|u - x_i\|^2 + (\bar{m}a_i + \kappa_0 m)\lambda D_h^2. \end{aligned}$$

Inequality (21) now follows by summing the above inequality from $i = 0$ to $i = k-1$ and rearranging terms. ∎

The following result develops a convergence rate bound for the quantity $\min_{1\le i\le k}\|v_i\|^2$. In view of the stopping criterion in step 3 of NC-FISTA, it plays a crucial role in establishing an iteration-complexity bound for NC-FISTA in Theorem 2.6.

Proposition 2.5 *Consider the sequences $\{y_k\}$ and $\{v_k\}$ generated by NC-FISTA according to* (10) *and* (12)*, respectively. Then, for every $k \ge 1$,*

$$v_k \in \nabla f(y_k) + \partial h(y_k) \tag{24}$$

and

$$\min_{1\le i\le k}\|v_i\|^2 \le \frac{4(2M + \kappa_0 m)^2}{M - \bar{M}}\left(\frac{\bar{m}D_h^2}{k} + \frac{3\kappa_0 m D_h^2}{k^2} + \frac{3\left[2A_0(\phi(y_0) - \phi_*) + (\kappa_0 m + M)d_0^2\right]}{k^3}\right) \tag{25}$$

where M, m, κ_0 and A_0 are as described in step 0 of NC-FISTA, D_h is defined in (5), $\bar{M}$ *and* $\bar{m}$ *are defined in the paragraph following assumptions (A1)-(A4), and*

$$d_0 := \inf_{z^* \in Z^*} \|z^* - y_0\| = \inf_{z^* \in Z^*} \|z^* - x_0\|. \tag{26}$$

Proof: The first conclusion (24) follows from the optimality condition of (10) and (12). Next we show the convergence rate bound (25) holds. First note that $A_0 > 0$ and the relation (8) with $k = 0$ imply that $a_0 > 1$. The assumptions that ∇f is $\bar{M}$-Lipschitz continuous (see (A3)), $M > \bar{M}$ and $\lambda = 1/M$ (see step 0 of NC-FISTA), relation (12) and the fact that $\{a_k\}$ is increasing then imply that

$$\min_{1\le i\le k} \|v_i\|^2 \le \left(\frac{1}{\lambda} + \frac{\kappa_0 m}{a_0} + \bar{M}\right)^2 \min_{0\le i\le k-1} \|y_{i+1} - \tilde{x}_i\|^2 \le (2M + \kappa_0 m)^2 \min_{0\le i\le k-1} \|y_{i+1} - \tilde{x}_i\|^2. \tag{27}$$

Moreover, due to the first remark after assumptions (A1)-(A4), there exists $z^* \in Z^*$ such that $\|z^* - x_0\| = d_0$. Noting that $z^* \in \operatorname{dom} h$, and using Lemma 2.4 with $u = z^*$, the fact that $\mathcal{C}_k \le \bar{M}$ for $k \ge 0$ and $\lambda = 1/M$, we conclude that

$$\begin{aligned}
\frac{M - \bar{M}}{M}\left(\sum_{i=0}^{k-1} A_{i+1}\right) \min_{0\le i\le k-1} \|y_{i+1} - \tilde{x}_i\|^2 &\le \sum_{i=0}^{k-1} \left((1 - \lambda\mathcal{C}_i) A_{i+1} \|y_{i+1} - \tilde{x}_i\|^2\right) \\
&\le 2\lambda A_0(\phi(y_0) - \phi_*) + (\kappa_0 m\lambda + 1)\, d_0^2 + \kappa_0 m\lambda D_h^2 k + \bar{m}\lambda D_h^2 \sum_{i=0}^{k-1} a_i \\
&= \frac{1}{M}\left[2A_0(\phi(y_0) - \phi_*) + (\kappa_0 m + M)\, d_0^2 + \kappa_0 m D_h^2 k + \bar{m} D_h^2 \sum_{i=0}^{k-1} a_i\right].
\end{aligned}$$

The bound (25) now follows by combining (27) with the above inequality and using Lemma A.1 in [18]. ∎

The following theorem presents the main result of this subsection. It describes an iteration-complexity bound for NC-FISTA involving both parameters M and m as described in its step 0.

Theorem 2.6 *Assume that the scalars M and A_0 in step 0 of NC-FISTA are such that*

$$\frac{M}{M - \bar{M}} = \mathcal{O}(1), \quad A_0 = \Theta(1). \tag{28}$$

Then, NC-FISTA outputs a $\hat{\rho}$-approximate solution $(\hat{y}, \hat{v})$ in at most

$$\mathcal{O}\left(\left(\frac{M\,(\phi(y_0) - \phi_*) + M^2 d_0^2}{\hat{\rho}^2}\right)^{1/3} + \left(\frac{M m D_h^2}{\hat{\rho}^2}\right)^{1/2} + \frac{M\bar{m}D_h^2}{\hat{\rho}^2} + 1\right) \tag{29}$$

iterations where m is as in step 0 of NC-FISTA, D_h is defined in (5), $\bar{m}$ *is defined in the paragraph following assumptions (A1)-(A4), and d_0 is defined in* (26).

Proof: Using the assumption that $A_0 = \Theta(1)$ and the definition of κ_0 in (7), we easily see that $\kappa_0 = \Theta(1)$. The iteration-complexity bound in (29) follows immediately from the second result in Proposition 2.5 (see (25)), (28), the stopping criterion in step 3 of NC-FISTA, and the facts that $M \ge m$ (see step 0 of NC-FISTA) and $\kappa_0 = \Theta(1)$. ∎

Note that if a sharper $m \in [\bar{m}, M]$ is not known and m is simply set to M, then (29) reduces to

$$\mathcal{O}\left(\left(\frac{M(\phi(y_0)-\phi_*)+M^2d_0^2}{\hat{\rho}^2}\right)^{1/3}+\frac{MD_h}{\hat{\rho}}+\frac{M\bar{m}D_h^2}{\hat{\rho}^2}+1\right).$$

Clearly, this special case only requires M as the AG method does and achieves the same iteration-complexity bound (in regards to the $\Theta(\hat{\rho}^{-2})$ dominant term).

3 An adaptive variant of the NC-FISTA

This section describes the second ACG variant studied in this paper, namely ADAP-NC-FISTA, which, in contrast to NC-FISTA, does not require the knowledge of a curvature pair (M, m) as input. Instead of choosing the parameters M and m as constants, it generates sequences $\{\mathcal{C}_k\}$ and $\{m_k\}$ (see (32), (33) and (34) below).

We begin by describing ADAP-NC-FISTA. Note that it requires as input an initial arbitrary pair (M_0, m_0) of positive scalars.

ADAP-NC-FISTA

---

0. Let an initial point $y_0 \in \operatorname{dom} h$, a scalar $\theta > 1$, a pair $(M_0, m_0) \in \mathbb{R}^2_{++}$ such that $M_0 \geq m_0$, and a tolerance $\hat{\rho} > 0$ be given, and set $x_0 = y_0$, $A_0 = 2$, $\lambda_0 = 1/M_0$ and $k = 0$;

1. compute a_k and A_{k+1} as in (8), $\tilde{x}_k$ as in (9),

$$\tilde{y}_k = \frac{A_k y_k + a_k y_0}{A_{k+1}}, \tag{30}$$

and

$$\underline{m}_{k+1} = \max\left\{\frac{2[\ell_f(\tilde{y}_k;\tilde{x}_k) - f(\tilde{y}_k)]}{\|\tilde{y}_k - \tilde{x}_k\|^2}, 0\right\}; \tag{31}$$

2. call the subroutine SUB(θ, λ_k, m_k) stated below to compute $(\lambda_{k+1}, m_{k+1}) = (\lambda, m)$ satisfying

$$\lambda \leq \lambda_k, \quad m \geq m_k, \tag{32}$$

$$\lambda C_k(\lambda, m) \leq 0.9, \tag{33}$$

$$2m\left(\lambda_k - \frac{\lambda}{a_k}\right) \geq \underline{m}_{k+1}\lambda, \tag{34}$$

where

$$C_k(\lambda, m) := \frac{2[f(y_k(\lambda, m)) - \ell_f(y_k(\lambda, m); \tilde{x}_k)]}{\|y_k(\lambda, m) - \tilde{x}_k\|^2}, \tag{35}$$

$$y_k(\lambda, m) := \operatorname{argmin}_u \left\{\ell_f(u; \tilde{x}_k) + h(u) + \frac{1}{2}\left(\frac{1}{\lambda} + \frac{2m}{a_k}\right)\|u - \tilde{x}_k\|^2\right\}, \tag{36}$$

and go to step 3;

3. compute

$$y_{k+1} = y_k(\lambda_{k+1}, m_{k+1}), \quad \mathcal{C}_{k+1} = C_k(\lambda_{k+1}, m_{k+1}), \tag{37}$$

$$x_{k+1} = P_\Omega\left(\frac{(a_k + 2m_{k+1}\lambda_{k+1})y_{k+1} - (a_k - 1)y_k}{2m_{k+1}\lambda_{k+1} + 1}\right),$$

$$v_{k+1} = \left(\frac{1}{\lambda_{k+1}} + \frac{2m_{k+1}}{a_k}\right)(\tilde{x}_k - y_{k+1}) + \nabla f(y_{k+1}) - \nabla f(\tilde{x}_k); \tag{38}$$

if $\|v_{k+1}\| \leq \hat{\rho}$ then output $(\hat{y}, \hat{v}) = (y_{k+1}, v_{k+1})$ and **stop**; otherwise, set $k \leftarrow k+1$ and go to step 1.

We will now describe the subroutine SUB(θ, λ, m) used in step 2 of ADAP-NC-FISTA to compute (λ, m) satisfying conditions (32)-(34).

SUB(θ, λ, m)

---

0. Compute $C_k(\lambda, m)$ and $y_k(\lambda, m)$ according to (35) and (36), respectively;

1. **if** (λ, m) satisfy both (33) and (34), then output (λ, m) and **stop**; **otherwise**, if (33) is not satisfied then set

$$\lambda^+ \leftarrow \min\left\{\frac{\lambda}{\theta}, \frac{0.9}{C_k(\lambda, m)}\right\}; \tag{39}$$

if (34) is not satisfied then set

$$m^+ \leftarrow 2m; \tag{40}$$

2. set $(\lambda, m) = (\lambda^+, m^+)$ and go to step 0.

We now make a few remarks about ADAP-NC-FISTA. First, ADAP-NC-FISTA consists of two types of iterations, namely, the ones indexed by k that we refer to as outer iterations and the ones performed inside SUB(θ, λ, m) that we refer to as inner iterations. Second, each inner iteration performs exactly one resolvent evaluation of h to compute $y_k(\lambda, m)$. Third, when the update (39) is performed, the quantity $C_k(\lambda, m)$ in the right hand side of (39) is always positive due to the fact that (33) is not satisfied and, as a consequence, λ^+ is well-defined and positive. Fourth, the choice of $A_0 = 2$, (8) with $k = 0$ and the fact that $\{a_k\}$ is increasing imply that $a_k \geq a_0 = 2$. Fifth, if f is convex, and hence $\bar{m} = 0$, and m_0 is set to 0 in ADAP-NC-FISTA, then it can be easily seen that the adaptive search for λ_k is equivalent to the adaptive search for the quantity L_k in [1] via the correspondence $L_k = 1/\lambda_k$. Thus, ADAP-NC-FISTA reduces to FISTA with backtracking when $\bar{m} = 0$.

The following lemma states some properties of ADAP-NC-FISTA.

Lemma 3.1 *The following statements hold for ADAP-NC-FISTA:*

(a) for every $k \geq 0$ and $\lambda, m > 0$, the quantities $C_k(\lambda, m)$ and $\mathcal{C}_{k+1}$ defined in (35) and (37), respectively, lie in $[-\bar{m}, \bar{M}]$;

(b) for every $k \geq 0$, the quantity $\underline{m}_{k+1}$ defined in (31) lies in $[0, \bar{m}]$;

(c) for every $k \geq 1$,

$$\mathcal{C}_k \lambda_k \leq 0.9, \quad 2m_k \lambda_{k-1} \geq \underline{m}_k \lambda_k + \frac{2m_k \lambda_k}{a_{k-1}};$$

(d) $\{\lambda_k\}$ is non-increasing and $\{m_k\}$ is non-decreasing;

(e) for every $k \geq 0$,

$$\lambda_k \geq \underline{\lambda} := \min\left\{\frac{0.9}{\theta \bar{M}}, \lambda_0\right\}, \qquad m_k \leq \max\{2\bar{m}, m_0\}; \tag{41}$$

Proof: (a)-(b) It follows from (2) (resp., (4)) and the fact that $\bar{M}$ (resp., $\bar{m}$) is the smallest scalar M (resp., m) satisfying (2) (resp., (4)) that $C_k(\lambda, m)$ and $\mathcal{C}_{k+1}$ (resp., $\underline{m}_{k+1}$) is bounded above by $\bar{M}$ (resp., $\bar{m}$). The quantities $C_k(\lambda, m)$ and $\mathcal{C}_{k+1}$ are bounded below by $-\bar{m}$ follows from $\bar{m}$ satisfying (4), and $\underline{m}_{k+1}$ is non-negative due to (31).

(c) The two conclusions follow from requirements (33) and (34).

(d) The requirements in (32) on (λ, m) immediately imply the two conclusions.

(e) We first prove the first inequality in (41). Indeed, assume for contradiction that it does not hold and let $\hat{k}$ be the smallest $k \geq 0$ such that $\lambda_k < \underline{\lambda}$. Since $\underline{\lambda} \leq \lambda_0$ in view of the definition of $\underline{\lambda}$ in (41), it follows from the definition of $\hat{k}$ that $\lambda_{\hat{k}}$ is obtained from (39), i.e.,

$$\lambda_{\hat{k}} = \lambda^+ := \min\left\{\frac{\lambda}{\theta}, \frac{0.9}{C_{\hat{k}-1}(\lambda, m)}\right\} \tag{42}$$

for some $(\lambda, m) \in (0, \lambda_0] \times \mathbb{R}_{++}$ such that (33) does not hold for the pair (λ, m) where $k = \hat{k} - 1$ in (33). Hence $C_{\hat{k}-1}(\lambda, m) > 0$ in view of the third remark following SUB(θ, λ, m). Moreover, it follows from the definition of $\underline{\lambda}$ in (41), statement (a) and the facts that $\theta > 1$ and $C_{\hat{k}-1}(\lambda, m) > 0$ that

$$\lambda_{\hat{k}} < \underline{\lambda} \leq \frac{0.9}{\theta \bar{M}} < \frac{0.9}{\bar{M}} \leq \frac{0.9}{C_{\hat{k}-1}(\lambda, m)}. \tag{43}$$

Clearly, (42) and (43) imply that $\lambda_{\hat{k}} = \lambda/\theta$. On the other hand, the fact that λ does not satisfy (33) and statement (a) imply that $\lambda > 0.9/C_{\hat{k}-1}(\lambda, m) \geq 0.9/\bar{M}$ and hence that $\lambda_{\hat{k}} = \lambda/\theta > 0.9/\theta\bar{M} \geq \underline{\lambda}$ due to the definition of $\underline{\lambda}$. Since the latter inequality contradicts our initial assumption, the first inequality in (41) follows. To prove the second inequality in (41), assume for contradiction that it does not hold and let $\bar{k} \geq 0$ be such that $m_{\bar{k}} > \max\{2\bar{m}, m_0\}$. It follows that $m_{\bar{k}} > m_0$ by the definition of $\hat{m}$ in (41), which, in view of (40), implies that $\bar{k} \geq 1$ and $m_{\bar{k}} = 2m$ for some $m \in \mathbb{R}_{++}$ that does not satisfied (34), i.e., m satisfies

$$2m\lambda_{\bar{k}-1} < \underline{m}_{\bar{k}} \lambda + \frac{2m\lambda}{a_{\bar{k}-1}}. \tag{44}$$

It then follows from (44), $\underline{m}_{\bar{k}} \leq \bar{m}$ due to statement (b), $\lambda \leq \lambda_{\bar{k}-1}$, and $a_{\bar{k}-1} \geq a_0 = 2$ that $m < \bar{m}$. The latter inequality and the fact that $m_{\bar{k}} = 2m$ imply that $m_{\bar{k}} < \max\{2\bar{m}, m_0\}$, which contradicts our initial assumption. Hence the second inequality in (41) follows. ∎

We have the following technical results that lead to Proposition 3.3, which then allows us to establish the iteration-complexity result for ADAP-NC-FISTA in Theorem 3.4.

Lemma 3.2 *For every $k \geq 0$ and $u \in \mathbb{R}^n$, we define*

$$\gamma_k(u) := \tilde{\gamma}_k(y_{k+1}) + \frac{1}{\lambda_{k+1}}\langle \tilde{x}_k - y_{k+1}, u - y_{k+1}\rangle + \frac{m_{k+1}}{a_k}\|u - y_{k+1}\|^2 \tag{45}$$

and

$$\tilde{\gamma}_k(u) := \ell_f(u; \tilde{x}_k) + h(u) + \frac{m_{k+1}}{a_k}\|u - \tilde{x}_k\|^2. \tag{46}$$

Then, for every $k \geq 0$, we have:

$$A_k\gamma_k(y_k) + a_k\gamma_k(y_0) \leq A_{k+1}\gamma_k(\tilde{y}_k) + m_{k+1}\|y_k - y_0\|^2, \tag{47}$$

$$A_{k+1}\phi(\tilde{y}_k) - A_k\phi(y_k) - a_k\phi(y_0) \leq \frac{\bar{m}a_k}{2}\|y_k - y_0\|^2, \tag{48}$$

$$\gamma_k(\tilde{y}_k) - \phi(\tilde{y}_k) \leq \frac{m_{k+1}\lambda_k}{A_{k+1}\lambda_{k+1}}\|y_0 - x_k\|^2. \tag{49}$$

Proof: Note that for any quadratic function $\gamma : \mathbb{R}^n \to \mathbb{R}$ with a quadratic term $\alpha\|\cdot\|^2$, every $A, a \in \mathbb{R}_+$ and $x, y \in \mathbb{R}^n$, we have

$$A\gamma(y) + a\gamma(x) = (A+a)\gamma\left(\frac{Ay + ax}{A+a}\right) + \frac{Aa}{A+a}\alpha\|y - x\|^2.$$

Applying the above identity with $\gamma = \gamma_k$, $A = A_k$, $a = a_k$, $y = y_k$ and $x = y_0$, and using the definition of $\tilde{y}_k$ in (30) and the relation (13), we obtain

$$A_k\gamma_k(y_k) + a_k\gamma_k(y_0) = A_{k+1}\gamma_k(\tilde{y}_k) + \frac{m_{k+1}A_k}{A_{k+1}}\|y_k - y_0\|^2 \leq A_{k+1}\gamma_k(\tilde{y}_k) + m_{k+1}\|y_k - y_0\|^2$$

where the inequality follows from the fact that $A_k \leq A_{k+1}$. Inequality (47) then follows. We now show (48). Due to the convexity of h, and relations (8) and (30), we have

$$A_{k+1}h(\tilde{y}_k) - A_kh(y_k) - a_kh(y_0) \leq 0.$$

It follows from $\phi = f + h$, the above inequality, the fact that $A_k \leq A_{k+1}$, and relations (4), (8) and (30) that

$$\begin{aligned} A_{k+1}\phi(\tilde{y}_k) - A_k\phi(y_k) - a_k\phi(y_0) &\leq A_{k+1}f(\tilde{y}_k) - A_kf(y_k) - a_kf(y_0) \\ &\leq \frac{\bar{m}A_ka_k}{2A_{k+1}}\|y_k - y_0\|^2 \leq \frac{\bar{m}a_k}{2}\|y_k - y_0\|^2. \end{aligned}$$

Next, we show (49). Using similar arguments as in the proof of Lemma 2.2(a), we have $\gamma_k(u) \leq \tilde{\gamma}_k(u)$ for every $u \in \operatorname{dom} h$. Hence, using (46), (31), (30), (9) and Lemma 3.1(c) that for every $k \geq 0$, we have

$$\begin{aligned} &\gamma_k(\tilde{y}_k) - \phi(\tilde{y}_k) \leq \tilde{\gamma}_k(\tilde{y}_k) - \phi(\tilde{y}_k) = \ell_f(\tilde{y}_k; \tilde{x}_k) - f(\tilde{y}_k) + \frac{m_{k+1}}{a_k}\|\tilde{y}_k - \tilde{x}_k\|^2 \\ &\leq \frac{1}{2}\left(\underline{m}_{k+1} + \frac{2m_{k+1}}{a_k}\right)\|\tilde{y}_k - \tilde{x}_k\|^2 = \frac{1}{2A_{k+1}}\left(\underline{m}_{k+1} + \frac{2m_{k+1}}{a_k}\right)\|y_0 - x_k\|^2 \leq \frac{m_{k+1}\lambda_k}{A_{k+1}\lambda_{k+1}}\|y_0 - x_k\|^2. \end{aligned}$$

Inequality (49) then follows. ∎

Proposition 3.3 *For every $k \geq 1$, we have*

$$\frac{1}{20}\left(\sum_{i=0}^{k-1}\frac{A_{i+1}}{m_{i+1}}\right)\min_{0\leq i\leq k-1}\|y_{i+1}-\tilde{x}_i\|^2 \leq \lambda_0 D_h^2\left(k+\bar{m}\sum_{i=0}^{k-1}\frac{a_i}{2m_{i+1}}\right)+\frac{2\lambda_0}{m_0}A_k(\phi(y_0)-\phi_*). \quad (50)$$

Proof: Using similar arguments as in the proof of Lemma 2.3 and the definition of $\mathcal{C}_i$ in (37), we conclude that for every $i \geq 0$ and $u \in \Omega$,

$$\begin{aligned}&2\lambda_{i+1}A_{i+1}\phi(y_{i+1})+(2m_{i+1}\lambda_{i+1}+1)\|u-x_{i+1}\|^2+(1-\lambda_{i+1}\mathcal{C}_{i+1})A_{i+1}\|y_{i+1}-\tilde{x}_i\|^2\\ &\leq 2\lambda_{i+1}A_i\gamma_i(y_i)+2\lambda_{i+1}a_i\gamma_i(u)+\|u-x_i\|^2, \end{aligned}\quad (51)$$

where γ_i and $\tilde{\gamma}_i$ are defined by (45) and (46), respectively. Using the relation (51) with $u = x_0$, Lemmas 3.2, 3.1(c)-(d), the facts that $x_0 = y_0$ and $\lambda_i \leq \lambda_0$ for $i \geq 0$, and the definition of D_h in (5) we conclude that for every $0 \leq i \leq k-1$,

$$\begin{aligned}&\frac{1}{10}A_{i+1}\|y_{i+1}-\tilde{x}_i\|^2+\left[2\lambda_{i+1}A_{i+1}(\phi(y_{i+1})-\phi(y_0))+(2m_{i+1}\lambda_{i+1}+1)\|x_0-x_{i+1}\|^2\right]\\ &\qquad-\left[2\lambda_{i+1}A_i(\phi(y_i)-\phi(y_0))+\|x_0-x_i\|^2\right]\\ &\leq 2\lambda_{i+1}A_i(\gamma_i(y_i)-\phi(y_i))+2\lambda_{i+1}a_i(\gamma_i(y_0)-\phi(y_0))\\ &= 2\lambda_{i+1}[A_i\gamma_i(y_i)+a_i\gamma_i(y_0)-A_{i+1}\phi(\tilde{y}_i)]+2\lambda_{i+1}[A_{i+1}\phi(\tilde{y}_i)-A_i\phi(y_i)-a_i\phi(y_0)]\\ &\leq 2\lambda_{i+1}\left[A_{i+1}(\gamma_i(\tilde{y}_i)-\phi(\tilde{y}_i))+m_{i+1}\|y_i-y_0\|^2\right]+\bar{m}a_i\lambda_{i+1}\|y_i-y_0\|^2\\ &\leq 2m_{i+1}\lambda_i\|y_0-x_i\|^2+2\lambda_{i+1}m_{i+1}\|y_i-y_0\|^2+\bar{m}a_i\lambda_{i+1}\|y_i-y_0\|^2\\ &\leq 2m_{i+1}\lambda_i\|x_0-x_i\|^2+(2m_{i+1}+\bar{m}a_i)\lambda_0 D_h^2\end{aligned}$$

where the second inequality follows from (47) and (48), the third inequality follows from (49). Dividing the above inequality by $2m_{i+1}$, rearranging terms and using the fact that, by Lemma 3.1(d), $m_i \leq m_{i+1}$, we obtain

$$\begin{aligned}\frac{A_{i+1}}{20m_{i+1}}\|y_{i+1}-\tilde{x}_i\|^2 \leq &\left[\frac{\lambda_i}{m_i}A_i(\phi(y_i)-\phi(y_0))+\left(\frac{1}{2m_i}+\lambda_i\right)\|x_0-x_i\|^2\right]\\ &-\left[\frac{\lambda_{i+1}}{m_{i+1}}A_{i+1}(\phi(y_{i+1})-\phi(y_0))+\left(\frac{1}{2m_{i+1}}+\lambda_{i+1}\right)\|x_0-x_{i+1}\|^2\right]\\ &+\left(\frac{\lambda_i}{m_i}-\frac{\lambda_{i+1}}{m_{i+1}}\right)A_i(\phi(y_0)-\phi(y_i))+\left(1+\frac{\bar{m}a_i}{2m_{i+1}}\right)\lambda_0 D_h^2.\end{aligned}$$

Summing the above inequality from $i = 0$ to $i = k-1$ and using the facts $\phi(y_i) \geq \phi_*$ for $i \geq 0$ and $\{\lambda_i/m_i\}$ is non-increasing due to Lemma 3.1(d), we obtain

$$\begin{aligned}&\frac{1}{20}\left(\sum_{i=0}^{k-1}\frac{A_{i+1}}{m_{i+1}}\right)\min_{0\leq i\leq k-1}\|y_{i+1}-\tilde{x}_i\|^2 \leq \frac{\lambda_k}{m_k}A_k(\phi(y_0)-\phi(y_k))-\left(\frac{1}{2m_k}+\lambda_k\right)\|x_0-x_k\|^2\\ &\quad+\sum_{i=0}^{k-1}\left(\frac{\lambda_i}{m_i}-\frac{\lambda_{i+1}}{m_{i+1}}\right)A_i(\phi(y_0)-\phi(y_i))+\lambda_0 D_h^2\left(k+\bar{m}\sum_{i=0}^{k-1}\frac{a_i}{2m_{i+1}}\right)\\ &\leq \frac{\lambda_0}{m_0}A_k(\phi(y_0)-\phi_*)+(\phi(y_0)-\phi_*)\sum_{i=0}^{k-1}\left(\frac{\lambda_i}{m_i}-\frac{\lambda_{i+1}}{m_{i+1}}\right)A_i+\lambda_0 D_h^2\left(k+\bar{m}\sum_{i=0}^{k-1}\frac{a_i}{2m_{i+1}}\right).\end{aligned}$$

Now, using the fact that $\{A_k\}$ is increasing and $\{\lambda_k/m_k\}$ is non-increasing, we have

$$\sum_{i=0}^{k-1}\left(\frac{\lambda_i}{m_i}-\frac{\lambda_{i+1}}{m_{i+1}}\right)A_i \le \frac{\lambda_0}{m_0}A_0+\sum_{i=1}^{k-1}(A_i-A_{i-1})\frac{\lambda_i}{m_i} \le \frac{\lambda_0}{m_0}A_0+\sum_{i=1}^{k-1}(A_i-A_{i-1})\frac{\lambda_0}{m_0} \le \frac{\lambda_0}{m_0}A_k.$$

Combining the above two inequalities, we then conclude that (50) holds. ∎

The next theorem is the main result of this section presenting the iteration-complexity for finding a $\hat{\rho}$-approximate solution of (1) by ADAP-NC-FISTA.

Theorem 3.4 *The following statements hold:*

(a) every iterate (y_k, v_k) generated by ADAP-NC-FISTA satisfies

$$v_k \in \nabla f(y_k)+\partial h(y_k);$$

moreover, ADAP-NC-FISTA outputs a $\hat{\rho}$-approximate solution $(\hat{y},\hat{v})$ in a number of outer iterations $\mathcal{T}$ bounded by

$$\mathcal{T}=\mathcal{O}\left(\left(\frac{C_1\bar{M}[\phi(y_0)-\phi_*]}{\hat{\rho}^2}\right)^{1/3}+\left(\frac{C_1\bar{M}m_0D_h^2}{\hat{\rho}^2}\right)^{1/2}+\frac{C_1\bar{M}\left[\bar{m}D_h^2+\phi(y_0)-\phi_*\right]}{\hat{\rho}^2}+1\right) \tag{52}$$

where D_h is defined in (5), $\bar{m}$ and $\bar{M}$ are defined in the paragraph following assumptions (A1)-(A4), and

$$C_1 := C_2\max\left\{\frac{\bar{m}}{m_0},1\right\}, \quad C_2 := \left[\sqrt{\frac{M_0}{\bar{M}}}+\sqrt{\frac{\bar{M}}{M_0}}\right]^2; \tag{53}$$

(b) if $m_0 \ge \bar{m}$, then an alternative bound on $\mathcal{T}$ is

$$\mathcal{T}=\mathcal{O}\left(\left(\frac{C_2\bar{M}\left[\phi(y_0)-\phi_*+M_0d_0^2\right]}{\hat{\rho}^2}\right)^{1/3}+\left(\frac{C_2\bar{M}m_0D_h^2}{\hat{\rho}^2}\right)^{1/2}+\frac{C_2\bar{M}\bar{m}D_h^2}{\hat{\rho}^2}+1\right), \tag{54}$$

where C_2 is defined in (53) and d_0 is defined in (26);

(c) the total number of inner iterations, and hence resolvent evaluations of h, performed by ADAP-NC-FISTA is bounded by

$$\mathcal{T}+\mathcal{O}\left(\log_1^+\left(\max\left\{\frac{\bar{M}}{M_0},\frac{\bar{m}}{m_0}\right\}\right)\right) \tag{55}$$

where $\log_1^+(\cdot)$ is defined in Subsection 1.1.

Proof: (a) The first conclusion follows from the same argument as in the proof of Proposition 2.5. Using the facts that $a_k \ge a_0 = 2$ from the fourth remark after SUB(θ,λ,m) and Lemma 3.1(e), we have

$$\frac{1}{\lambda_{k+1}}+\frac{2m_{k+1}}{a_k} \le \frac{1}{\underline{\lambda}}+\max\{2\bar{m},m_0\}$$

for every $k \geq 0$. This conclusion together with the definition of $\bar{M}$ in the paragraph following assumptions (A1)-(A4), assumption (A3) and (38) then implies that

$$\begin{aligned}\min_{1\leq i\leq k}\|v_i\| &\leq \min_{0\leq i\leq k-1}\left(\frac{1}{\lambda_{i+1}}+\frac{2m_{i+1}}{a_i}+\bar{M}\right)\|y_{i+1}-\tilde{x}_i\| \\ &\leq \left(\frac{1}{\underline{\lambda}}+\max\{2\bar{m},m_0\}+\bar{M}\right)\min_{0\leq i\leq k-1}\|y_{i+1}-\tilde{x}_i\|. \end{aligned} \tag{56}$$

Moreover, using the definition of $\underline{\lambda}$ in (41), the facts that $\bar{m}\leq\bar{M}$ and $\lambda_0 = 1/M_0$, and the definition of C_1 in (53), we have

$$\frac{1}{\underline{\lambda}}+\max\{2\bar{m},m_0\}+\bar{M} \leq \left(\frac{\theta}{0.9}+3\right)\left(M_0+\bar{M}\right) \leq (2\theta+5)\sqrt{\frac{C_1\bar{M}M_0m_0}{\max\{2\bar{m},m_0\}}}.$$

Using Proposition 3.3, Lemma 3.1 (d)-(e), the above two inequalities, the fact that $A_k = A_0 + \sum_{i=0}^{k-1} a_i$ due to (8), and rearranging terms, we obtain

$$\begin{aligned}&\frac{1}{20}\left(\sum_{i=0}^{k-1}A_{i+1}\right)\min_{1\leq i\leq k}\|v_i\|^2 \\ &\leq (2\theta+5)^2C_1\bar{M}\left[2A_0(\phi(y_0)-\phi_*)+m_0D_h^2k+\left[\frac{\bar{m}D_h^2}{2}+2(\phi(y_0)-\phi_*)\right]\sum_{i=0}^{k-1}a_i\right].\end{aligned}$$

The complexity bound (52) now follows immediately from the above inequality and Lemma A.1 in [18].

(b) The proof of this statement is similar to the proof of (a) except that Proposition A.1 is used in place of Proposition 3.3.

(c) It suffices to argue that the total number of times that the pair (λ, m) is updated inside all calls to the subroutine SUB(θ, λ, m) is bounded by the second term in (55). Indeed, this assertion follows from the following facts: the initial value of (λ, m) is (λ_0, m_0) (see step 0 of ADAP-NC-FISTA); in view of (33) and (34), the pair (λ, m) is no longer updated whenever $\lambda \leq 0.9/\bar{M}$ and $m \geq 2\bar{m}$, and; due to (39) and (40), λ is reduced by a factor less than or equal to $\theta > 1$ and m is increased by a factor of 2 each time either one of them is updated. ∎

We now make two remarks about ADAP-NC-FISTA in light of NC-FISTA. First, in contrast to NC-FISTA, the input pair (M_0, m_0) of ADAP-NC-FISTA can be an arbitrary pair in $\mathbb{R}^2_{++}$. Second, if (M, m) denotes a pair as in step 0 of NC-FISTA, then it can be easily seen that $(M_0, m_0) = (M, m)$ satisfies the assumption of Theorem 3.4(b) and the complexity bound (54) for ADAP-NC-FISTA with input pair $(M_0, m_0) = (M, m)$ reduces to the complexity bound (29) for NC-FISTA.

We end this section by making a few final remarks about the iteration-complexity bound derived in Theorem 3.4(b) for the case in which $M_0 = \mathcal{O}(\bar{M})$. First, in this case, the dominant term of the complexity bound (54) is $\mathcal{O}\left(\bar{M}^2\bar{m}D_h^2/(M_0\hat{\rho}^2)\right)$, and hence it increases as M_0 decreases. Second, the best choice of M_0 that minimizes the constant C_2 in (53) is $M_0 = \Theta(\bar{M})$. However, computational experiments indicate that taking smaller values for M_0 improves the performance of the method. One reason that may explain this phenomenon is that the constant $\bar{M}$ that appears in (56), and as a consequence in C_1, C_2, and the other terms that appear in the bounds (52) and (54), is very conservative and close examination of the proof of Theorem 3.4 shows that it can actually be replaced by the sharper (and potentially smaller) quantity

$$L_k := \frac{\|\nabla f(y_{\hat{k}}) - \nabla f(\tilde{x}_{\hat{k}-1})\|}{\|y_{\hat{k}} - \tilde{x}_{\hat{k}-1}\|},$$

where $\hat{k} = \operatorname{argmin}_i\{\|y_i - \tilde{x}_{i-1}\| : 1 \leq i \leq k\}$.

4 Computational results

This section reports experimental results obtained by our implementation of NC-FISTA, ADAP-NC-FISTA, and three variants of the latter method, on four problems that are instances of the SNCO problem (1), namely: nonconvex quadratic programming problem in both vector (Subsection 4.1) and matrix versions (Subsection 4.2), matrix completion (Subsection 4.3) and nonnegative matrix factorization (NMF, Subsection 4.4). Note that NMF is a problem for which $\operatorname{dom} h$ is unbounded.

We start by describing the three variants of ADAP-NC-FISTA considered in our computational benchmark, namely, R-ADAP-NC-FISTA, ADAP-NC-FISTA-BB and R-ADAP-NC-FISTA-BB. The first one is a restart variant of ADAP-NC-FISTA, namely, it restarts the latter method with input $y_0 = y_k$ and $(M_0, m_0) = (M_0, m_k)$ whenever $\phi(y_{k+1}) \geq \phi(y_k)$ (hence, without resetting k to 0, this is equivalent to rejecting y_{k+1} and setting $x_k = y_k$, $A_k = A_0$ and $\lambda_k = \lambda_0$). The last two variants are heuristic variants of ADAP-NC-FISTA and R-ADAP-NC-FISTA, respectively, which invokes in step 2 the subroutine SUB with input $(\theta, \tilde{\lambda}_k, m_k)$ where

$$\tilde{\lambda}_k = \begin{cases} \lambda_k^{BB} := \frac{\langle s_{k-1}, g_{k-1}\rangle}{\|g_{k-1}\|^2}, & \text{if } \lambda_k^{BB} > 0; \\ \frac{1}{M_0}, & \text{otherwise} \end{cases}$$

where $s_{k-1} = \tilde{x}_{k-1} - y_k$ and $g_{k-1} = \nabla f(\tilde{x}_{k-1}) - \nabla f(y_k)$.

For the sake of simplicity, we use the abbreviations NC, AD, AD(B), RA and RA(B) to refer to NC-FISTA, ADAP-NC-FISTA, ADAP-NC-FISTA-BB, R-ADAP-NC-FISTA and R-ADAP-NC-FISTA-BB, respectively, both in the discussions and tables below. The triples (M, m, A_0) and (M_0, m_0, θ) which are used as input for NC and AD, respectively, depend on the problem under consideration and are described in the four subsections below. Moreover, AD(B), RA and RA(B) use the same input triple as AD.

We compare our methods with four others: the AG method proposed in [6], the NM-APG method proposed in [16], and the UPFAG and UPFAG-BB methods proposed in [7]. Note that all four methods are natural extensions of ACG variants for solving convex programs to the context of nonconvex optimization problems. For the sake of simplicity, we use the abbreviations NM, UP and UP(B) to refer to NM-APG, UPFAG and UPFAG-BB, respectively, both in the discussions and tables below.

We now provide the details of our implementation of the four methods mentioned in the previous paragraph. AG was implemented as described in Algorithm 1 of [6] with sequences $\{\alpha_k\}$, $\{\beta_k\}$ and $\{\lambda_k\}$ chosen as $(\alpha_k, \beta_k, \lambda_k) = (2/(k+1), 0.99/M, k\beta_k/2)$ for $k \geq 1$. NM was implemented as described in Algorithm 2 of [16] with the quadruple $(\alpha_x, \alpha_y, \eta, \delta)$ chosen to be $(0.99/M, 0.99/M, 0.9, 1)$. The code for UP was made available by the authors of [7] where UP is described (see Algorithm 1 of [7]). In particular, we have used their choice of parameters but have modified the code slightly to accommodate for the termination criterion (6) used in our benchmark. More specifically, the parameters $(\hat{\lambda}_0, \hat{\beta}_0, \gamma_1, \gamma_2, \gamma_3, \delta)$ needed as input by UP were set to $(1/\bar{M}, 1/\bar{M}, 0.4, 0.4, 1, 10^{-3})$. UP(B) also requires the same parameters as UP and an additional one denoted by σ in [7] which were set to the same values used in UP and to $\sigma = 10^{-10}$, respectively.

It is worth making the following remarks about the above method: i) AG and NM require two resolvent evaluations of h per iteration while NC requires only one (see the third remark after NC);

ii) NM reduces to the composite gradient method when a certain descent property is not satisfied; iii) AD, AD(B), RA, RA(B), UP and UP(B) can work without the knowledge of a curvature pair (M, m); and iv) UP and UP(B) adaptively compute both accelerated steps and unaccelerated ones using line searches.

We implement all methods in MATLAB 2017b scripts and run them on a MacBook Pro with a 4-core Intel Core i7 processor and 16 GB of memory.

4.1 Nonconvex quadratic programming problem

This subsection discusses the performance of NC and its adaptive variants to solve the same quadratic programming problem as in [14, 18], namely:

$$\min \left\{ f(z) := -\frac{\alpha_1}{2}\|DBz\|^2 + \frac{\alpha_2}{2}\|Az - b\|^2 : z \in \Delta_n \right\}, \tag{57}$$

where $(\alpha_1, \alpha_2) \in \mathbb{R}^2_{++}$, $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with diagonal entries sampled from the discrete uniform distribution $\mathcal{U}\{1, 1000\}$, matrices $A \in \mathbb{R}^{l\times n}$, $B \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^l$ are such that their entries are generated from the uniform distribution $\mathcal{U}[0, 1]$, and $\Delta_n := \{z \in \mathbb{R}^n : \sum_{i=1}^n z_i = 1, \ \ z_i \geq 0\}$ is the $(n-1)$-dimensional standard simplex. The dimensions are set to be $(l, n) = (20, 1200)$. For some chosen curvature pairs $(\bar{m}, \bar{M}) \in \mathbb{R}^2_{++}$, the scalars α_1 and α_2 were chosen so that $\bar{M} = \lambda_{\max}(\nabla^2 f)$ and $-\bar{m} = \lambda_{\min}(\nabla^2 f)$ where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the largest and smallest eigenvalues functions, respectively. Note that we set $\Omega = \mathbb{R}^n$ in this subsection.

In addition to the nine methods described at the beginning of Section 4, this subsection (and only this one) also reports the performance of a quasi-Newton variant of UPFAG, called QN, as described in [7] (see its paragraph containing (2.13)). Each iteration of QN performs an unaccelerated step with respect to a variable metric and whose computation requires the evaluation of a point-to-point operator of the form $(I + V^{-1}\partial h)^{-1}(\cdot)$ for some $V \in S^n_{++}$ (see [2]). More specifically, QN is almost the same as UP (and hence has the same set of parameters as UP), except that it replaces (2.10) by (2.13) in [7], where the quasi-Newton matrix G_k in (2.13) is updated as in the symmetric-rank-1 method (see [2]).

In our implementation, all methods use the centroid of Δ_n as the initial point z_0 and terminate with a pair (z, v) satisfying

$$v \in \nabla f(z) + N_{\Delta_n}(z), \qquad \frac{\|v\|}{\|\nabla f(z_0)\| + 1} \leq 10^{-7}. \tag{58}$$

The input triple of NC is set to $(M, m, A_0) = (\bar{M}/0.99, \bar{m}, 1000)$ and that of AD is set to $(M_0, m_0, \theta) = (1, 1, 1.25)$.

Test cases specified by pairs $(\bar{M}, \bar{m})$ are generated by choosing the corresponding α_1 and α_2 as discussed in the first paragraph in this subsection. Computational results for ten methods with fixed $\bar{M} = 16777216$ are presented Table 1 and with fixed $\bar{m} = 1$ are presented in Table 2. In each table, the first column gives the values of $\bar{m}$ or $\bar{M}$ used to generate the instances, the second to eighth (resp., ninth to eleventh) columns provide the number of iterations and running times of AG, UP, QN, NM, NC, AD and RA (resp., UP(B), AD(B) and RA(B)). The objective function values obtained by all methods are not reported since they are essentially the same on all instances. The bold numbers highlight the methods (using and without using Barzilai-Borwein stepsizes) that have the best performance for each case. The numbers marked with * indicate that the maximum number of iterations has been reached.

$\bar{m}$	Iteration Count / Running Time (s)							Iteration Count / Running Time (s)		
	AG	UP	QN	NM	NC	AD	RA	UP(B)	AD(B)	RA(B)
16777216	638	220	219	251	2376	3	3	605	3	3
	97	47	64	31	286	**1**	**1**	258	3	**2**
1048576	1358	1176	103	1157	3469	318	58	10	19	17
	224	252	34	184	421	63	**12**	**6**	9	**6**
65536	22293	5676	2737	44705	3832	747	80	30	57	30
	3524	1284	959	6525	459	157	**18**	16	20	**10**
4096	31385	8286	919	50000*	17585	1000	74	39	90	36
	5184	1918	320	7070	2101	211	**18**	21	34	**14**
256	26961	7464	3410	49602	31333	969	76	35	95	44
	4369	1667	1126	7001	3713	216	**18**	18	34	**17**
16	26918	7334	665	49515	32517	967	75	30	80	34
	4215	1609	221	6806	3958	223	**18**	15	29	**13**

Table 1: Numerical results for instances with fixed $\bar{M} = 16777216$

$\bar{M}$	Iteration Count / Running Time (s)							Iteration Count / Running Time (s)		
	AG	UP	QN	NM	NC	AD	RA	UP(B)	AD(B)	RA(B)
4000	31403	7857	50000*	50000*	17577	244	105	43	58	58
	5284	1682	16214	7270	2244	50	**20**	**15**	18	17
16000	20193	7857	50000*	50000*	30239	472	79	35	51	34
	3504	1739	14850	7884	3638	105	**18**	15	18	**12**
64000	26962	7464	50000*	49592	31334	560	77	38	64	37
	4891	1652	15511	7628	3803	125	**18**	16	23	**13**
256000	26926	7364	3488	49534	32527	930	75	38	72	36
	4759	1522	1131	7541	3980	206	**18**	20	27	**14**
1024000	26918	7364	3234	49521	32518	967	74	38	77	35
	4717	1601	1028	7815	4092	227	**18**	22	29	**13**
4096000	26916	7264	99	49523	32515	967	79	39	82	36
	4547	1602	33	7847	4265	231	**18**	21	32	**13**

Table 2: Numerical results for instances with fixed $\bar{m} = 1$

In summary, computational results demonstrate that: i) among the methods which do not use the Barzilai-Borwein stepsize (see columns 2-8 of Tables 1-2), RA has the best performance in terms of running time; ii) UP(B) is comparable with RA (see columns 8 and 9 of Tables 1-2); and iii) RA(B) has the best performance among the three methods which use the Barzilai-Borwein stepsize (see columns 9-11 of Tables 1-2).

4.2 Matrix problem

In this subsection, we test our methods on a matrix version of the nonconvex quadratic programming problem

$$\min\left\{f(Z) := -\frac{\alpha_1}{2}\|D\mathcal{B}(Z)\|^2 + \frac{\alpha_2}{2}\|\mathcal{A}(Z) - b\|^2 : Z \in P_n\right\},$$

where $\mathcal{A} : S_+^n \to \mathbb{R}^l$ and $\mathcal{B} : S_+^n \to \mathbb{R}^n$ are linear operators defined by

$$[\mathcal{A}(Z)]_i = \langle A_i, Z\rangle_F \text{ for } A_i \in \mathbb{R}^{n\times n} \text{ and } 1 \le i \le l,$$
$$[\mathcal{B}(Z)]_j = \langle B_j, Z\rangle_F \text{ for } B_j \in \mathbb{R}^{n\times n} \text{ and } 1 \le j \le n,$$

with entries of A_i, B_j sampled from the uniform distribution $\mathcal{U}[0,1]$, and P_n denotes the spectraplex

$$P_n := \{Z \in S_+^n : \mathrm{tr}(Z) = 1\}.$$

(α_1, α_2), D and b are defined as those in Subsection 4.1. Note that we set $\Omega = S_+^n$ in this subsection.

All methods used the centroid of P_n as the initial point Z_0, i.e., $Z_0 = I_n/n$, where I_n is the identity matrix of size $n \times n$. Termination criterion is the same as (58) except that Δ_n is replaced by P_n. The input triple of NC is set to $(M, m, A_0) = (\bar{M}/0.99, \bar{m}, 1000)$ and that of AD is set to $(M_0, m_0, \theta) = (1, 1000, 1.25)$.

Test cases specified by pairs $(\bar{M}, \bar{m})$ are generated by choosing the corresponding α_1 and α_2 as discussed in the first paragraph in this subsection. Computational results of all methods with fixed $\bar{M} = 1000000$ are presented in Tables 3-5. Their formats are the same as that of Table 1. The objective function values obtained by all methods are not reported since they are essentially the same on all instances. The bold numbers highlight the methods (using and without using Barzilai-Borwein stepsizes) that have the best performance for each case.

$\bar{m}$	Iteration Count / Running Time (s)						Iteration Count / Running Time (s)		
	AG	UP	NM	NC	AD	RA	UP(B)	AD(B)	RA(B)
1000000	46	12	80	33	12	12	9	11	12
	2	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**
100000	3809	2577	6242	3960	2206	597	2573	593	282
	138	113	191	94	87	**25**	274	41	**21**
10000	5400	7697	10404	1247	2591	1290	6811	835	569
	198	347	328	**29**	103	54	671	57	**40**
1000	4621	6759	11053	4424	2637	1211	6384	721	581
	163	308	360	111	104	**51**	646	48	**41**
100	4476	6620	11271	8870	2639	1373	6876	812	535
	157	299	312	218	113	**57**	683	54	**37**

Table 3: Numerical results for instances with fixed $\bar{M} = 1000000$

In Table 3, the dimensions are set to be $(l, n) = (50, 200)$ and 2.5% of entries in A_i, B_j are nonzero.

$\bar{m}$	Iteration Count / Running Time (s)						Iteration Count / Running Time (s)		
	AG	UP	NM	NC	AD	RA	UP(B)	AD(B)	RA(B)
1000000	44	12	75	32	12	12	10	12	12
	4	**1**	5	2	2	2	**2**	**2**	**2**
100000	1411	621	3151	635	530	240	57	151	61
	134	69	224	40	52	**25**	13	28	**11**
10000	1963	1733	5071	1104	868	198	109	211	137
	195	191	373	69	86	**21**	31	39	**25**
1000	1935	1792	5172	3823	900	215	97	208	160
	193	197	382	244	94	**23**	**25**	38	29
100	1934	1803	5045	5771	904	210	112	225	147
	190	197	367	391	95	**23**	29	40	**27**

Table 4: Numerical results for instances with fixed $\bar{M} = 1000000$

In Table 4, the dimensions are set to be $(l, n) = (50, 400)$ and 0.5% of entries in A_i, B_j are nonzero.

$\bar{m}$	Iteration Count / Running Time (s)						Iteration Count / Running Time (s)		
	AG	UP	NM	NC	AD	RA	UP(B)	AD(B)	RA(B)
1000000	69	16	117	39	11	11	13	11	11
	22	6	26	8	**5**	6	8	**7**	**7**
100000	277	58	502	165	24	8	9	8	8
	119	21	118	39	10	**3**	7	**4**	**4**
10000	491	141	1030	703	60	60	13	13	13
	173	52	246	168	23	**21**	10	**7**	8
1000	531	161	1144	1326	70	70	13	15	15
	169	60	259	309	26	**25**	10	**9**	**9**
100	535	163	1156	1482	71	71	13	16	16
	172	61	260	336	26	**25**	**10**	**10**	**10**

Table 5: Numerical results for instances with fixed $\bar{M} = 1000000$

In Table 5, the dimensions are set to be $(l, n) = (50, 800)$ and 0.1% of entries in A_i, B_j are nonzero.

In summary, computational results demonstrate that: i) among the methods which do not use the Barzilai-Borwein stepsize (see columns 2-7 of Tables 3-5), RA has the best performance in terms of running time; ii) UP(B) is comparable with RA in many instances (see columns 7 and 8 of Tables 3-5); and iii) RA(B) has the best performance among the three methods which use the Barzilai-Borwein stepsize (see columns 8-10 of Tables 3-5).

4.3 Matrix completion

This subsection focuses on a constrained version of the nonconvex low-rank matrix completion problem studied in [19, 28].

Given an incomplete observed matrix O with the set $\mathcal{Q}$ of observed entries, parameters $\beta > 0$

and $\tau > 0$ and letting $p : \mathbb{R} \to \mathbb{R}_+$ denote the log-sum penalty

$$p(t) = p_{\beta,\tau}(t) := \beta \log\left(1 + \frac{|t|}{\tau}\right)$$

and $\Pi_{\mathcal{Q}}$ denote the linear operator that maps a matrix A to the matrix whose entries in $\mathcal{Q}$ have the same values of the corresponding ones in A and whose entries outside of $\mathcal{Q}$ are all zero, then the constrained version of the matrix completion problem is formulated as

$$\min_{X \in \mathbb{R}^{l\times n}} f(X) + h(X), \tag{59}$$

where

$$f(X) = \frac{1}{2}\|\Pi_{\mathcal{Q}}(X - O)\|_F^2 + \mu \sum_{i=1}^{r} [p(\sigma_i(X)) - p_0\sigma_i(X)],$$
$$h(X) = \mu p_0 \|X\|_* + I_{\mathcal{B}(R)}(X), \quad p_0 = p'(0) = \frac{\beta}{\tau},$$

R is a positive scalar, $\mathcal{B}(R) := \{X \in \mathbb{R}^{l\times n} : \|X\|_F \le R\}$, $O \in \mathbb{R}^{\mathcal{Q}}$ is an incomplete observed matrix, $\mu > 0$ is a parameter, $r := \min\{l, n\}$ and $\sigma_i(X)$ is the i-th singular value of X and $\|\cdot\|_*$ denotes the nuclear norm defined as $\|\cdot\|_* := \sum_{i=1}^{r} \sigma_i(\cdot)$. Note that we set $\Omega = \mathbb{R}^{l\times n}$ in this subsection. It is shown in [19, 28] that the problem in (59) falls into the general class of SNCO problems,

$$f(X') - f(X) - \langle \nabla f(X'), X' - X\rangle_F \le \frac{\tilde{M}}{2}\|X' - X\|_F^2, \quad \forall X, X' \in \Omega$$

for $\tilde{M} = 1$ and that the pair

$$(\bar{M}, \bar{m}) = \left(\max\left\{\tilde{M}, \frac{2\mu\beta}{\tau^2}\right\}, \frac{2\mu\beta}{\tau^2}\right) \tag{60}$$

satisfies (2) and (4).

We use the *MovieLens* dataset[1] to obtain the observed index set $\mathcal{Q}$ and the incomplete observed matrix O. The dataset includes a sparse matrix with 100,000 ratings of {1,2,3,4,5} from 943 users on 1682 movies. The radius R is chosen as the Frobenius norm of the matrix of size 943×1682 containing the same entries as O in $\mathcal{Q}$ and 5 in the entries outside of $\mathcal{Q}$.

All methods take a random matrix Z_0 sampled from the standard Gaussian distribution as the initial point, where the random number generation seed is fixed, and terminates with a pair (Z, V) satisfying

$$V \in \nabla f(Z) + \partial h(Z), \qquad \frac{\|V\|_F}{\|\nabla f(Z_0)\|_F + 1} \le 5 \times 10^{-4}.$$

The input triple of NC is set to $(M, m, A_0) = (\tilde{M}, \tilde{M}, 2)$, since $\tilde{M}$ is the one actually needed in the convergence analysis of this algorithm (see Lemma 2.4). The input triple of AD is set to $(M_0, m_0, \theta) = (1, 0.5, 1.25)$.

Computational results of all methods are summarized in Table 6. Specifically, the first column gives the values of $\bar{M}$ computed according to (60) with four different triples (μ, β, τ), the second to seventh columns provide the function values of (59) at the last iteration and the number of

[1] http://grouplens.org/datasets/movielens/

iterations, and the eighth to thirteenth columns present the running times. The bold numbers highlight the methods that have the best performance for each case. The results of RA are not reported since they are the same as those of AD, which is due to the fact that $\{\phi(y_k)\}$ generated by AD is a decreasing sequence and hence no restart is performed in RA.

$\bar{M}$	Function Value / Iteration Count						Running Time (s)					
	AG	UP	UP(B)	NM	NC	AD	AG	UP	UP(B)	NM	NC	AD
4.4	2257	2670	2605	**1809**	2605	2625	4568	2214	1545	1033	1114	**1021**
	3856	898	521	1036	1491	1219						
8.9	3886	4322	4261	**3359**	4154	4203	10251	2592	1621	1605	1202	**1089**
	9158	1782	576	1617	1642	1302						
20	4282	4736	4637	**3635**	4637	4582	29274	5850	1914	2836	**1178**	1822
	22902	3962	898	2875	676	2177						
30	5967	6475	6753	**5237**	6292	6293	41673	8159	1628	4182	**1233**	1633
	37032	5857	606	3717	1646	1952						

Table 6: Numerical results for matrix completion instances

In summary, computational results demonstrate that: i) NM always finds the smallest function values, since it requires objective function values to satisfy a descent property, and if violated, a projected gradient step is taken to ensure the descent in function values; ii) NC and AD have the best performance in terms of the running time; and iii) since NC and AD are good enough compared with UP(B), we do not presents the results of AD(B) and RA(B).

4.4 Nonnegative matrix factorization

In this subsection, we further test AD on a real life application rather than artificially generated problems and data. NMF is a popular dimension reduction method in which a data matrix X is factored into two matrices V and W, with constraints that each entry in V and W is nonnegative.

$$\min \left\{ f(V,W) := \frac{1}{2}\|X - VW\|_F^2 : V \geq 0, W \geq 0 \right\}, \tag{61}$$

where $X \in \mathbb{R}^{n\times l}$, $V \in \mathbb{R}^{n\times k}$ and $W \in \mathbb{R}^{k\times l}$. Note that we set $\Omega = \mathbb{R}^{n\times l}$ in this subsection. Intuitively, the data matrix X is a collection of m data points in $\mathbb{R}^n$, the columns of V can be viewed as the basis of all data points, and hence each data point is a linear combination of the basis, with weights in the corresponding column in W. Because of its ability of extracting easily interpretable factors and automatically performing clustering, NMF finds a wide range of applications in practice, from text mining to image processing. Most of the NMF algorithms solve (61) in a two-block coordinate descent manner, by alternatively minimizing with respect to one of the two blocks, V or W, while keeping the other one fixed. Alternating minimization is a natural idea for NMF, since the subproblem in one block is convex.

In this subsection, we apply AD to solve the nonconvex problem (61) directly by minimizing in (V, W) jointly.

For a preliminary computational test, we apply AD to facial feature extraction. The problem is as described in (61), to factor out a data matrix into two matrices. The facial image dataset is provided by AT&T Laboratories Cambridge [2]. There are ten different images of each of 40 distinct

[2] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

subjects, and each image contains 92×112 pixels, with 256 gray levels per pixel. It results in a matrix of size $10,304 \times 400$, where each column of the data matrix is the vectorization of an image.

It is hard to estimate M in (2) due the unboundedness in NMF, so we can only apply AD, which has the benefit of working without the knowledge of M. AD is benchmarked against the ANLS (Alternating Nonnegative Least Squares) method [12]. ANLS alternatively solves minimization subproblems in V and W with nonnegative constraints and the other variable being fixed. We use the implementation of ANLS [3] provided by the authors of [12] as a benchmark for comparison. The ANLS code is slightly modified to accommodate for the termination criterion (62).

Both methods use the initial point $(V_0, W_0) = (\mathbb{1}^{n\times k}/(nk), \mathbb{1}^{k\times l}/(kl))$, where $\mathbb{1}^{n\times k}$ and $\mathbb{1}^{k\times l}$ are all one matrices of size $n\times k$ and $k\times l$. k is set to be 20. AD terminates with a pair $((V, W), (S_V, S_W))$ satisfying

$$(S_V, S_W) \in \nabla f(V, W) + N_{\mathcal{F}}(V, W), \qquad \frac{\|(S_V, S_W)\|_F}{\|\nabla f(V_0, W_0)\|_F + 1} \leq 10^{-7}, \tag{62}$$

where $\mathcal{F} = \{(V, W) \in \mathbb{R}^{n\times k} \times \mathbb{R}^{k\times l} : V \geq 0, W \geq 0\}$. The input triple of AD is set to $(M_0, m_0, \theta) = (1, 1000, 1.25)$. Computational results are summarized in Table 7.

Method	Function Value	Iteration Count	Running Time(s)
AD	2.80E+09	28	4.6
ANLS	1.20E+09	1000*	137.6

Table 7: Numerical results for NMF

In summary, computational results demonstrate that ANLS reaches the maximum number of iterations (i.e., 1000), and AD outperforms ANLS in terms of the running time.

5 Concluding remarks

This paper presents two ACG variants and establishes their iteration-complexities for obtaining an approximate solution of the SNCO problem. Numerical results are also given showing that they are both efficient in practice.

We have not assumed in our analysis that the set Ω as in assumption (A3) is bounded. However, we remark that if Ω is bounded then it can be shown using a simpler analysis than the one given in this paper that the version of the NC-FISTA with $m = 0$ and $\lambda = 1/(2M)$ has an

$$\mathcal{O}\left(\left(\frac{M^2 d_0^2}{\hat{\rho}^2}\right)^{1/3} + \left(\frac{M\bar{m}D_\Omega^2}{\hat{\rho}^2}\right)^{1/2} + \frac{M\bar{m}D_h^2}{\hat{\rho}^2} + 1\right)$$

iteration-complexity where $D_\Omega := \sup_{u,u'\in\Omega} \|u' - u\| < \infty$. Moreover, it can be shown that a version of the ADAP-NC-FISTA in which λ_k is updated in a similar way and $m_k = 0$ for every k has a guaranteed iteration-complexity that lies in between the one above and the one in (52).

Finally, we have implemented the two versions mentioned in the previous paragraph and tested them on problems for which Ω is bounded but have observed that they are not as efficient as the corresponding ones studied in this paper.

[3] https://www.cc.gatech.edu/ hpark/nmfsoftware.html

6 Acknowledgements

We are grateful to Guanghui Lan and Saeed Ghadimi for sharing the source code of the UPFAG method in [7]. We are also grateful to the two anonymous referees and the associate editor for their insightful comments which we have used to substantially improve the quality of this work.

References

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

[2] S. Becker and J. Fadili. A quasi-newton proximal splitting method. In *Advances in Neural Information Processing Systems*, volume 25, pages 2618–2626, 2012.

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

[4] Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

[5] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, Jul 2018.

[6] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156:59–99, 2016.

[7] S. Ghadimi, G. Lan, and H. Zhang. Generalized uniformly optimal methods for nonlinear programming. *Journal of Scientific Computing*, 79(3):1854–1881, 2019.

[8] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *international conference on machine learning*, pages 37–45. PMLR, 2013.

[9] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[10] 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 Journal on Optimization*, 25:2182–2211, 2015.

[11] Y. He and R. D. C. Monteiro. An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM Journal on Optimization*, 26:29–56, 2016.

[12] J. Kim and H. Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *2008 Eighth IEEE International Conference on Data Mining*, pages 353–362. IEEE, 2008.

[13] O. Kolossoski and R. D. C. Monteiro. An accelerated non-Euclidean hybrid proximal extragradient-type algorithm for convex-concave saddle-point problems. *Optimization Methods and Software*, 32:1244–1272, 2017.

[14] 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 Journal on Optimization*, 29(4):2566–2593, 2019.

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

[16] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 379–387, December 2015.

[17] Q. Li, Y. Zhou, Y. Liang, and P. K. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *International Conference on Machine Learning*, pages 2111–2119. PMLR, 2017.

[18] J. Liang and R. D. C. Monteiro. A doubly accelerated inexact proximal point method for nonconvex composite optimization problems. *Available on arXiv:1811.11378*, 2018.

[19] J. Liang and R. 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.

[20] R. D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM J. Optim.*, 23(2):1092–1125, 2013.

[21] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $\mathrm{O}(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983.

[22] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140:125–161, 2013.

[23] Y. E. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[24] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr. An accelerated linearized alternating direction method of multipliers. *SIAM J. Imaging Sci.*, 8(1):644–681, 2015.

[25] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of Machine Learning Research: International Conference on Artificial Intelligence and Statistics*, volume 84, pages 613–622, April 2018.

[26] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[27] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. `http://www.mit.edu/~dimitrib/PTseng/papers.html`, 2008.

[28] Q. Yao and J. T. Kwok. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. *Journal of Machine Learning Research*, 18:179–1, 2017.

[29] Q. Yao, J. T. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3308–3314. IJCAI, 2017.

A Supplementary results

This section provides a bound on the quantity $\min_{0\leq i\leq k-1}\|y_{i+1}-\tilde{x}_i\|^2$ for the case in which the parameter m_0 of the ADAP-NC-FISTA satisfies $m_0\geq\bar{m}$. Note that an alternative bound on this quantity has already been developed in Proposition 3.3 for any $m_0>0$.

Proposition A.1 *For every $k\geq 1$, for $m_0\geq\bar{m}$, we have*

$$\frac{1}{10}\left(\sum_{i=0}^{k-1}A_{i+1}\right)\min_{0\leq i\leq k-1}\|y_{i+1}-\tilde{x}_i\|^2\leq 2\lambda_0A_0(\phi(y_0)-\phi_*)+\|x_0-x^*\|^2+\lambda_0D_h^2\left(2m_0+2m_0k+\bar{m}\sum_{i=0}^{k-1}a_i\right).$$

Proof: Using the assumption of the lemma that $m_0\geq\bar{m}$, the facts that $a_i\geq 2$ for $i\geq 0$ from the fourth remark following SUB(θ,λ,m), and $\{\lambda_i\}$ is non-increasing from Lemma 3.1(d), we have

$$\left(\bar{m}+\frac{2m_0}{a_i}\right)\lambda_{i+1}\leq m_0\left(1+\frac{2}{a_i}\right)\lambda_{i+1}\leq 2m_0\lambda_i. \tag{63}$$

The above inequality implies that (34) is always satisfied with $m=m_0$ and $\lambda=\lambda_{i+1}$. Hence, m_k is never updated in SUB(θ,λ,m), i.e., $m_i=m_0$, for $i\geq 0$. Using similar arguments as in the proof of Lemma 2.3, we conclude that for every $i\geq 0$ and $u\in\Omega$,

$$\begin{aligned}&2\lambda_{i+1}A_{i+1}\phi(y_{i+1})+(2m_0\lambda_{i+1}+1)\|u-x_{i+1}\|^2+(1-\lambda_{i+1}\mathcal{C}_{i+1})A_{i+1}\|y_{i+1}-\tilde{x}_i\|^2\\&\leq 2\lambda_{i+1}A_i\gamma_i(y_i)+2\lambda_{i+1}a_i\gamma_i(u)+\|u-x_i\|^2,\end{aligned} \tag{64}$$

where

$$\gamma_i(u):=\tilde{\gamma}_i(y_{i+1})+\frac{1}{\lambda_{i+1}}\langle\tilde{x}_i-y_{i+1},u-y_{i+1}\rangle+\frac{m_0}{a_i}\|u-y_{i+1}\|^2$$

and

$$\tilde{\gamma}_i(u):=\ell_f(u;\tilde{x}_i)+h(u)+\frac{m_0}{a_i}\|u-\tilde{x}_i\|^2. \tag{65}$$

As in Lemma 2.2(a), we have $\gamma_i(u)\leq\tilde{\gamma}_i(u)$ for every $u\in\operatorname{dom}h$. Hence, it follows from (65) and (4) that for every $k\geq 0$ and $u\in\operatorname{dom}h$, we have

$$\gamma_i(u)-\phi(u)\leq\tilde{\gamma}_i(u)-\phi(u)=\ell_f(u;\tilde{x}_i)-f(u)+\frac{m_0}{a_i}\|u-\tilde{x}_i\|^2\leq\frac{1}{2}\left(\bar{m}+\frac{2m_0}{a_i}\right)\|u-\tilde{x}_i\|^2. \tag{66}$$

Taking $u=x^*$, and using (64), (23), (66), (63), Lemma 3.1(c), and the facts that $x_0=y_0$, $\lambda_i\leq\lambda_0$ and $\phi(y_i)\geq\phi_*$ for $i\geq 0$, we conclude that for every $0\leq i\leq k-1$,

$$\begin{aligned}0.1&A_{i+1}\|y_{i+1}-\tilde{x}_i\|^2-2\lambda_iA_i(\phi(y_i)-\phi_*)-\|x^*-x_i\|^2\\&+2\lambda_{i+1}A_{i+1}(\phi(y_{i+1})-\phi_*)+(2m_0\lambda_{i+1}+1)\|x^*-x_{i+1}\|^2\\&\leq 2\lambda_{i+1}A_i(\gamma_i(y_i)-\phi(y_i))+2\lambda_{i+1}a_i(\gamma_i(x^*)-\phi_*)+2(\lambda_{i+1}-\lambda_i)A_i(\phi(y_i)-\phi_*)\\&\leq\lambda_{i+1}\left(\bar{m}+\frac{2m_0}{a_i}\right)\left(A_i\|y_i-\tilde{x}_i\|^2+a_i\|x^*-\tilde{x}_i\|^2\right)\end{aligned}$$

$$
\begin{aligned}
&\leq \lambda_{i+1}\left(\bar{m}+\frac{2m_0}{a_i}\right)\left(\|x^*-x_i\|^2+a_iD_h^2\right)\\
&\leq 2m_0\lambda_i\|x_i-x^*\|^2+(\bar{m}a_i+2m_0)\lambda_{i+1}D_h^2\\
&\leq 2m_0\lambda_i\|x_i-x^*\|^2+(\bar{m}a_i+2m_0)\lambda_0D_h^2.
\end{aligned}
$$

The conclusion is obtained by rearranging terms and summing the above inequality from $i=0$ to $k-1$. ∎