Accelerated Inexact Composite Gradient Methods for Nonconvex Spectral Optimization Problems*

Weiwei Kong † Renato D.C. Monteiro‡

July 22, 2020 (Revised: July 7, 2021 and February 9, 2022)

Abstract

This paper presents two inexact composite gradient methods, one inner accelerated and another doubly accelerated, for solving a class of nonconvex spectral composite optimization problems. More specifically, the objective function for these problems is of the form $f_1 + f_2 + h$, where f_1 and f_2 are differentiable nonconvex matrix functions with Lipschitz continuous gradients, h is a proper closed convex matrix function, and both f_2 and h can be expressed as functions that operate on the singular values of their inputs. The methods essentially use an accelerated composite gradient method to solve a sequence of proximal subproblems involving the linear approximation of f_1 and the singular value functions underlying f_2 and h. Unlike other composite gradient-based methods, the proposed methods take advantage of both the composite and spectral structure underlying the objective function in order to efficiently generate their solutions. Numerical experiments are presented to demonstrate the practicality of these methods on a set of real-world and randomly generated spectral optimization problems.

Keywords: composite nonconvex problem, iteration complexity, inexact composite gradient method, first-order accelerated gradient method, spectral optimization.

1 Introduction

There are numerous applications in electrical engineering, machine learning, and medical imaging that can be formulated as nonconvex spectral optimization problems of the form

$$\min_{U\in\mathbb{R}^{m\times n}} \left\{ \phi(U) := f_1(U) + (\underbrace{f_2^{\mathcal{V}} \circ \sigma}_{f_2})(U) + (\underbrace{h^{\mathcal{V}} \circ \sigma}_{h})(U) \right\}, \tag{1}$$

where σ is the function that maps a matrix to its singular value vector (in nonincreasing order of magnitude), f_1 and $f_2^{\mathcal{V}}$ are continuously differentiable functions with Lipschitz continuous gradients, and $h^{\mathcal{V}}$ is a proper, lower semicontinuous, convex function. For this paper, we are interested in

*The works of these authors were partially supported by ONR Grant N00014-18-1-2077, AFOSR Grant FA9550-22-1-0088, NSERC Grant PGSD3-516700-2018, and the IDEaS-TRIAD Fellowship (NSF Grant CCF-1740776). The first author has been supported by the US Department of Energy (DOE) and UT-Battelle, LLC, under contract DE-AC05-00OR22725 and also supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, 37830 (E-mail: wwkong92@gmail.com).

‡School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (E-mail: monteiro@isye.gatech.edu).

solving instances of (1) where: (i) the resolvents of $\lambda\partial h$ and $\lambda\partial h^{\mathcal{V}}$, i.e., evaluations of the operators $(I+\lambda\partial h)^{-1}$ and $(I+\lambda\partial h^{\mathcal{V}})^{-1}$, are easy compute for any $\lambda>0$; (ii) the resolvents of $\lambda(\nabla f_2+\partial h)$ and $\lambda(\nabla f_2^{\mathcal{V}}+\partial h^{\mathcal{V}})$ cannot be computed exactly for any $\lambda>0$; and (iii) both $f_2^{\mathcal{V}}$ and $h^{\mathcal{V}}$ are absolutely symmetric in their arguments, i.e., they do not depend on the ordering or the sign of their arguments.

We now describe some practical instances of (1) that satisfy all three assumptions above. To avoid repetition, we let $\mathcal{R}=\mathcal{R}_s+\mathcal{R}_n$ and $\mathcal{P}$ be two sparsity-inducing regularizers, where $\mathcal{R}_s$ and $\mathcal{P}$ are continuously differentiable functions with Lipschitz continuous gradients and $\mathcal{R}_n$ is a proper, lower semicontinuous, and convex function.

- *Matrix Completion.* Let $A\in\mathbb{R}^{m\times n}$ be a given data matrix and let $r=\min\{m,n\}$. Moreover, let Ω denote a subset of the indices of A. The goal of the general matrix completion problem is to find a low rank approximation of A that is close to A in some sense. A nonconvex formulation (see, for example, [21]) of this problem is

$$\min_{X\in\mathbb{R}^{m\times n}}\left\{\frac{1}{2}\|P_\Omega(X-A)\|_F^2+(\mathcal{R}\circ\sigma)(X)\right\},$$

  where P_Ω is the function that zeros out the entries of its input that are not in Ω. Note that this problem is a special instance of (1) in which $f_1=\|P_\Omega(\cdot)-A\|_F^2/2$, $f_2^{\mathcal{V}}=\mathcal{R}_s$, and $h^{\mathcal{V}}=\mathcal{R}_n$.

- *Phase Retrieval.* Given a vector $x\in\mathbb{R}^n$, let $x[\omega]$ denote its discrete Fourier transform for some frequency ω. Moreover, for some unknown noisy signal $\tilde{x}\in\mathbb{R}^n$ and a frequency set $\Omega\subseteq\mathbb{R}_+$, suppose that we are given measurements $\{|\tilde{x}[\omega]|\}_{\omega\in\Omega}$ and vectors $a_\omega\in\mathbb{C}^n$ such that $|\langle a_\omega,\tilde{x}\rangle|=|\tilde{x}[\omega]|$ for every $\omega\in\Omega$. The goal of the phase retrieval problem is to recover an approximation x of $\tilde{x}$ such that $|\langle a_\omega,x\rangle|^2\approx|\langle a_\omega,\tilde{x}\rangle|^2$ for every $\omega\in\Omega$. A nonconvex formulation of this problem is

$$\min_{X\in\mathbb{R}^{|\Omega|\times|\Omega|}}\left\{\frac{1}{2}\|\mathcal{A}(X)-b\|^2+(\mathcal{R}\circ\lambda)(X):X\succeq 0\right\},$$

  where λ denotes the function that maps matrices to their eigenvalue vector, $X\succeq 0$ means that X is symmetric positive semidefinite, and the quantities $\mathcal{A}:\mathbb{R}^{|\Omega|\times|\Omega|}\mapsto\mathbb{R}^{|\Omega|}$ and $b\in\mathbb{R}^{|\Omega|}$ are given by

$$[\mathcal{A}(X)]_\omega=\operatorname{tr}(a_\omega a_\omega^* X),\quad b_\omega=|\tilde{x}[\omega]|^2,\quad\forall(X,\omega)\in\mathbb{R}^{|\Omega|\times|\Omega|}\times\Omega.$$

  Note that this problem is a special instance of (1) in which $f_1=\|\mathcal{A}(\cdot)-b\|_F^2/2$, $f_2^{\mathcal{V}}=\mathcal{R}_s$, and $h^{\mathcal{V}}=\mathcal{R}_n+\delta_{\mathbb{R}_+^{|\Omega|}}$ where $\delta_{\mathbb{R}_+^{|\Omega|}}$ is the indicator for the nonnegative orthant of $\mathbb{R}^{|\Omega|}$. It is worth mentioning that this formulation is a generalization of the one in [3] where the convex function $\operatorname{tr}X$ is replaced with the nonconvex function $\mathcal{R}$.

- *Robust Principal Component Analysis.* Let $\widehat{M}\in\mathbb{R}^{m\times n}$ be a given data matrix and let $r=\min\{m,n\}$. The goal of the robust principal component analysis problem is to find an approximation $M+E$ of $\widehat{M}$ where M is low-rank and E is sparse. A nonconvex formulation of this problem is

$$\min_{M,E\in\mathbb{R}^{m\times n}}\left\{\frac{1}{2}\|\widehat{M}-(M+E)\|_F^2+(\mathcal{R}\circ\sigma)(M)+\mathcal{P}(E)\right\}.$$

  Note that this problem is a special instance of (1) in which $f_1=\|\widehat{M}-[(\cdot)-E]\|_F^2/2+\mathcal{P}$, $f_2^{\mathcal{V}}=\mathcal{R}_s$, and $h^{\mathcal{V}}=\mathcal{R}_n$. It is worth mentioning that this formulation is a instance of the one in [20] where more structure is imposed on the functions $\mathcal{R}$ and $\mathcal{P}$.

A natural approach for finding approximate stationary points of the above instances is to employ the *exact* composite gradient (ECG) method that, when applied to (1), *exactly* solves a sequence of matrix subproblems of the form

$$\min_{U\in\mathbb{R}^{m\times n}} \left\{\tilde{\lambda}_k\left[\langle\nabla(f_1+f_2)(Y_{k-1}),U\rangle+h(U)\right]+\frac{1}{2}\|U-Y_{k-1}\|_F^2\right\}, \tag{2}$$

where $\lambda_k>0$ is an appropriately chosen stepsize and the point Y_{k-1} is the previous iterate. Its computation primarily consists of computing a singular value decomposition (SVD) at the point $\tilde{Y}_k:=Y_{k-1}-\tilde{\lambda}_k\nabla(f_1+f_2)(Y_{k-1})$ and an evaluation of the resolvent of $\tilde{\lambda}_k\partial h^{\mathcal{V}}$ at $\sigma(\tilde{Y}_k)$. Accelerated ECG (A-ECG) methods solve subproblems similar to (2) but with Y_{k-1} selected in an accelerated manner. Notice that both of these approaches do not exploit the spectral structure in f_2.

Our goal in this paper is to develop two efficient *inexact* composite gradient (ICG) methods that find approximate stationary points of (1) by exploiting the spectral structure in *both* f_2 and h. Our first prototype, called the static inner accelerated ICG (IA-ICG) method, *inexactly* solves a sequence of matrix prox subproblems of the form

$$\min_{U\in\mathbb{R}^{m\times n}} \left\{\lambda_k\left[\langle\nabla f_1(Y_{k-1}),U\rangle+f_2(U)+h(U)\right]+\frac{1}{2}\|U-Y_{k-1}\|_F^2\right\} \tag{3}$$

where $\lambda_k>0$ is an appropriately chosen stepsize and the point Y_{k-1} is the previous iterate. It is shown (see Subsection 4.1) that the effort of finding the required inexact solution Y_k of (3) consists of computing one SVD and applying an accelerated gradient (ACG) algorithm to find an approximate solution to the related vector prox subproblem

$$\min_{u\in\mathbb{R}^r}\left\{\lambda_k\left[f_2^{\mathcal{V}}(u)-\langle c_{k-1},u\rangle+h^{\mathcal{V}}(u)\right]+\frac{1}{2}\|u\|^2\right\} \tag{4}$$

where $r=\min\{m,n\}$ and $c_{k-1}=\sigma(Y_{k-1}-\lambda_k\nabla f_1(Y_{k-1}))$. Notice that (4) is a problem over the vector space $\mathbb{R}^r$, and hence, has significantly fewer dimensions than (3) which is a problem over the matrix space $\mathbb{R}^{m\times n}$. The other prototype, called the static doubly accelerated ICG (DA-ICG), solves a subproblem similar to (3) but with Y_{k-1} selected in an accelerated manner (and hence its qualifier "doubly accelerated"). Notice that the static IA-ICG (resp. DA-ICG) can be viewed as an inexact version of ECG (resp. A-ECG) where, instead of h in (2), the function f_2+h is viewed as the composite term, i.e., the part that is not linearized in the subproblems. Moreover, neither IA-ICG nor DA-ICG are able to solve (3) (or its accelerated version) exactly due to assumption (ii) made in the first paragraph of this section.

Motivation of our approach. For high-dimensional instances of (1) where $r=\min\{m,n\}$ is large, we have that the larger the Lipschitz constant of $\nabla f_2^{\mathcal{V}}$ is, the better the performance of the ICG methods is compared to the performance of their exact counterparts. This fact immediately follows from the following two claims:

(i) the ICG methods inexactly solve fewer matrix subproblems compared to their exact counterparts when the Lipschitz constant of $\nabla f_2^{\mathcal{V}}$ is large; and

(ii) the work of exactly solving (2) or inexactly solving (3) is comparable when r is large.

The justification of claim (i) is as follows. First, recall that the larger the stepizes λ_k's (resp. $\tilde{\lambda}_k$) are, the smaller the number of generated subproblems (3) (resp. (2)) is. Second, the CG stepsizes chosen in either (2) or (3) to guarantee convergence of the underlying CG method are inversely

proportional to the Lipschitz constant of the gradient of the function being linearized. Hence, since the inexact CG methods linearize f_1 only and the exact CG methods linearize both f_1 and f_2, claim (i) follows. Some specific applications where the Lipschitz constant of $\nabla f_2^{\mathcal{Y}}$ may be large in practice can be found, for example, in [1, 19, 21]. The justification for claim (ii) is due to the following two observations: (a) all of the above CG methods require one SVD per subproblem; and (b) when r is large, the computational bottleneck for solving a single subproblem is the aforementioned SVD.

Contributions and Main results. To the best of our knowledge, this paper is the first to present ICG methods that exploit both the spectral and composite structure in (1).

When f_2 is convex or, more generally, a key inequality is satisfied at every iteration of ACG applied to (4), it is shown that for any given $\hat{\rho} > 0$, both the static IA-ICG and the static DA-ICG always obtain a pair $(\hat{Y}, \hat{V})$ satisfying the approximate stationarity condition

$$\hat{V} \in \nabla f_1(\hat{Y}) + \nabla f_2(\hat{Y}) + \partial h(\hat{Y}), \quad \|\hat{V}\| \leq \hat{\rho}. \tag{5}$$

by inexactly solving $\mathcal{O}(\hat{\rho}^{-2})$ matrix prox subproblems as in (3). If, in addition, f_1 is convex, it is shown that this bound improves to $\mathcal{O}(\hat{\rho}^{-2/3})$ for the static DA-ICG method.

When f_2 is nonconvex, the static IA-ICG and the static DA-ICG may fail to obtain a pair as in (5). To remedy this, we develop dynamic IA-ICG and DA-ICG methods that repeatedly invoke their static counterparts to solve (1) with (f_1, f_2) replaced by $(f_{1,\xi}, f_{2,\xi}) = (f_1 - \xi\|\cdot\|^2/2, f_2 + \xi\|\cdot\|^2/2)$ for strictly increasing values of $\xi > 0$. These dynamic versions always obtain a pair as in (5) because: (i) $f_1 + f_2 = f_{1,\xi} + f_{2,\xi}$ for every $\xi > 0$ and (ii) there always exists $\underline{\xi} > 0$ such that $f_{2,\underline{\xi}}$ is convex due to the fact that ∇f_2 is Lipschitz continuous.

Numerical experiments are also given to demonstrate the practicality of our proposed methods. More specifically, our experiments demonstrate that the dynamic methods are substantially faster (usually 10x) than other first-order methods at minimizing the primal residual $\|\hat{V}\|$ in terms of runtime.

Related works. The earliest complexity analysis of an ACG method for solving nonconvex composite problems like the one in (1) is given in [6]. Building on the results in [6], many other papers [5, 7, 13] have proposed similar ACG-based methods.

Another common approach for solving problems like (1) is to employ an inexact proximal point method where each prox subproblem is constructed to be convex, and hence, solvable by an ACG variant. For example, papers [4, 9, 10, 17] present inner accelerated inexact proximal point methods whereas [12] presents a doubly accelerated inexact proximal point method.

Organization of the paper. Subsection 1.1 gives some notation and basic definitions. Section 2 presents some necessary background material for describing the ICG methods. Section 3 is split into three subsections. The first one precisely describes the problem of interest, while the last two present the IA-ICG and DA-ICG methods. Section 4 describes an efficient way of solving problem (3) by modifying a solution of problem (4). Section 5 presents some numerical results. Section 6 establishes the iteration complexity of the ICG methods. Finally, some auxiliary results are presented in Appendices A to D.

1.1 Notation and Basic Definitions

This subsection provides some basic notation and definitions.

The set of real numbers is denoted by $\mathbb{R}$. The set of non-negative real numbers and the set of positive real numbers is denoted by $\mathbb{R}_+$ and $\mathbb{R}_{++}$ respectively. The set of natural numbers is

denoted by $\mathbb{N}$. The set of complex numbers is $\mathbb{C}$. The set of unitary matrices of size n–by–n is $\mathcal{U}^n$. For $t > 0$, define $\log_1^+(t) := \max\{1, \log(t)\}$. Let $\mathbb{R}^n$ denote a real–valued n–dimensional Euclidean space with norm $\|\cdot\|$. Given a linear operator $A : \mathbb{R}^n \mapsto \mathbb{R}^p$, the operator norm of A is denoted by $\|A\| := \sup\{\|Az\|/\|z\| : z \in \mathbb{R}^n, z \neq 0\}$. Using the asymptotic notation $\mathcal{O}$, we denote $\mathcal{O}_1(\cdot) \equiv \mathcal{O}(1 + \cdot)$.

Let $(m, n) \in \mathbb{N}^2$ and let $r = \min\{m, n\}$. Given matrices $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{n\times n}$, let the quantities $\sigma(X)$ and $\lambda(Y)$ denote the singular values and eigenvalues of X and Y, respectively, in nonincreasing order. Let $\operatorname{dg} : \mathbb{R}^r \mapsto \mathbb{R}^{r\times r}$ and $\operatorname{Dg} : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^r$ be given pointwise by

$$[\operatorname{dg} z]_{ij} = \begin{cases} z_i, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases} \qquad [\operatorname{Dg} Z]_i = Z_{ii},$$

for every $z \in \mathbb{R}^r$, $Z \in \mathbb{R}^{m\times n}$, and $(i, j) \in \{1, ..., r\}^2$.

The following notation and definitions are for a general complete inner product space $\mathcal{Z}$, whose inner product and its associated induced norm are denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ respectively. Let $\psi : \mathcal{Z} \mapsto (-\infty, \infty]$ be given. The effective domain of ψ is denoted by $\operatorname{dom}\psi := \{x \in \mathcal{Z} : \psi(x) < \infty\}$ and ψ is said to be proper if $\operatorname{dom}\psi \neq \emptyset$. For $\varepsilon \geq 0$, the ε-subdifferential of ψ at $x \in \operatorname{dom}\psi$ is denoted by

$$\partial_\varepsilon\psi(z) := \left\{w \in \mathbb{R}^n : \psi(z') \geq \psi(z) + \langle w, z' - z\rangle - \varepsilon, \forall z' \in \mathcal{Z}\right\},$$

and we denote $\partial\psi \equiv \partial_0\psi$. The set of proper, lower semi-continuous, convex functions is denoted by $\overline{\text{Conv}}\ \mathcal{Z}$. The convex conjugate ψ is denoted by ψ^*. The linear approximation of ψ at a point $z_0 \in \operatorname{dom}\psi$ is denoted by $\ell_\psi(\cdot; z_0) := \psi(z_0) + \langle\nabla\psi(z_0), \cdot - z_0\rangle$. The indicator of a closed convex set $C \subseteq \mathcal{Z}$ at a point $z \in \mathcal{Z}$ is denoted by $\delta_C(z)$, which is 1 if $z \in C$ and ∞ otherwise. The local Lipschitz constant of $\nabla\psi$ at two points $u, z \in \mathcal{Z}$ is denoted by

$$L_\psi(x, y) = \begin{cases} \frac{\|\nabla\psi(x)-\nabla\psi(y)\|}{\|x-y\|}, & x \neq y, \\ 0, & x = y, \end{cases} \quad \forall x, y \in \operatorname{dom}\psi. \tag{6}$$

2 Background Material

Recall from Section 1 that our interest is in solving (1) by repeatedly solving a sequence of prox subproblems as in (3). This section presents some background material regarding (3).

This section considers the nonconvex composite optimization (NCO) problem

$$\min_{u\in\mathcal{Z}} \left\{\psi(u) := \psi_s(u) + \psi_n(u)\right\}, \tag{7}$$

where $\mathcal{Z}$ is a finite dimensional inner product space and the functions ψ_s and ψ_n are assumed to satisfy the following assumptions:

(B1) $\psi_n \in \overline{\text{Conv}}\ \mathcal{Z}$;

(B2) ψ_s is continuously differentiable on $\mathcal{Z}$ and satisfies $\psi_s(u) - \ell_{\psi_s}(u; y) \leq M\|u - y\|^2/2$ for some $M \geq 0$ and every $u, y \in \mathcal{Z}$.

Clearly, problems (1) and (3) are special cases of (7), and hence any definition or result that is stated in the context of (7) applies to (1) and/or (3).

An important notion of an approximate solution of (7) is as follows: given $\hat{\rho} > 0$, a pair (y_r, v_r) is said to be a $\hat{\rho}$–approximate solution of (7) if

$$v_r \in \nabla\psi_s(y_r) + \partial\psi_n(y_r), \quad \|v_r\| \leq \hat{\rho}. \tag{8}$$

In Section 3, we develop prox-type methods for finding $\hat{\rho}$–approximate solutions of (1) that repeatedly solve (3) inexactly by taking advantage of its spectral decomposition.

We now discuss the inexactness criterion under which the subproblems (3) are solved. Again, the criterion is described in the context of (7) as follows.

Problem $\mathcal{A}$: Given $(\mu, \theta) \in \mathbb{R}^2_{++}$ and $z_0 \in \mathcal{Z}$, find $(y, v, \varepsilon) \in \operatorname{dom}\psi \times \mathcal{Z} \times \mathbb{R}_+$ such that

$$v \in \partial_\varepsilon \left(\psi - \frac{\mu}{2}\|\cdot - y\|^2\right)(y), \quad \|v\|^2 + 2\varepsilon \leq \theta^2\|y - z_0\|^2. \tag{9}$$

We begin by making three remarks about the above problem. First, if (y, v, ε) solves Problem $\mathcal{A}$ with $\theta = 0$, then $(v, \varepsilon) = (0, 0)$, and z is an exact solution of (7). Hence, the output (y, v, ε) of Problem $\mathcal{A}$ can be viewed as an inexact solution of (7) when $\theta \in \mathbb{R}_{++}$. Second, the input z_0 is arbitrary for the purpose of this section. However, the two methods described in Section 3 for solving (1) repeatedly solve (3) according to Problem $\mathcal{A}$ with the input z_0 at the k^{th} iteration determined by the iterates generated at the $(k-1)^{\text{th}}$ iteration. Third, defining the function

$$\Delta_\mu(u; y, v) := \psi(y) - \psi(u) - \langle v, y - u\rangle + \frac{\mu}{2}\|u - y\|^2 \quad \forall u \in \operatorname{dom}\psi, \tag{10}$$

another way to express the inclusion in (9) is $\Delta_\mu(u; y, v) \leq \varepsilon$ for every $u \in \operatorname{dom}\psi$. Finally, the relaxed ACG (R-ACG) algorithm presented later in this subsection will be shown to solve Problem $\mathcal{A}$ when ψ_s is convex. Moreover, it solves a weaker version of Problem $\mathcal{A}$ involving Δ_μ (see Problem $\mathcal{B}$ later on) whenever ψ_s is not convex and as long as some key inequalities are satisfied during its execution.

A technical issue in our analysis in this paper lies in the ability of refining the output of Problem $\mathcal{A}$ to an approximate solution (y_r, v_r) of (7), i.e., one satisfying the inclusion in (8), in which $\|v_r\|$ is nicely bounded. We now present a refinement procedure that addresses this issue.

Refinement Procedure

Input: a triple (M, ψ_s, ψ_n) satisfying (B1)–(B2) and a pair $(y, v) \in \operatorname{dom}\psi_n \times \mathcal{Z}$;
Output: a pair (y_r, v_r) satisfying the inclusion in (8);

1. set the quantities

$$y_r = \operatorname*{argmin}_{u \in \mathcal{Z}} \left\{\langle \nabla\psi_s(y) - v, u\rangle + \frac{M}{2}\|u - y\|^2 + \psi_n(u)\right\}, \tag{11}$$

$$v_r = v + M(y - y_r) + \nabla\psi_s(y_r) - \nabla\psi_s(y), \tag{12}$$

and output (y_r, v_r).

The result below presents the key properties of the above procedure. For the sake of brevity, we write $(y_r, v_r) = RP(y, v)$ to indicate that the pair (y_r, v_r) is the output of the above procedure with inputs (M, ψ_s, ψ_n) and (y, v).

Proposition 1. *Let (M, ψ_s, ψ_n) satisfying assumptions (B1)–(B2) and a triple $(y, v, \varepsilon) \in \operatorname{dom}\psi_n \times \mathcal{Z} \times \mathbb{R}_+$ be given. Moreover, let $(y_r, v_r) = RP(y, v)$, denote $L_{\psi_s}(\cdot, \cdot)$ simply by $L(\cdot, \cdot)$ where $L_{\psi_s}(\cdot, \cdot)$ is as in (6), and let Δ_μ be as in (10). Then, the following statements hold:*

(a) $v_r \in \nabla\psi_s(y_r) + \partial\psi_n(y_r)$;

(b) $\Delta_\mu(y_r; y, v) \geq M\|y_r - y\|^2/2$;

(c) *if* $\Delta_\mu(y_r; y, v) \leq \varepsilon$ *and* (y, v, ε) *satisfies the inequality in* (9)*, then*

$$\|v_r\| \leq \theta\left[1 + \frac{M + L(y, y_r)}{\sqrt{M}}\right]\|y - z_0\|; \tag{13}$$

(d) *if* (y, v, ε) *solves Problem* $\mathcal{A}$*, then* $\Delta_\mu(u; y, v) \leq \varepsilon$ *for every* $u \in \operatorname{dom}\psi_n$*, and, as a consequence, bound* (13) *holds.*

Proof. (a) Using the definition of v_r and the optimality of y_r, we have that

$$v_r = v + M(y - y_r) + \nabla\psi_s(y_r) - \nabla\psi_s(y) \in \nabla\psi_s(y_r) + \partial\psi_n(y_r).$$

(b) Let $(y, v) \in \operatorname{dom}\psi_n \times \mathcal{Z}$ be fixed, and define $\widetilde{\psi}_s := \psi_s - \langle v, \cdot\rangle$. Using Proposition 19 with $(g, h, L) = (\widetilde{\psi}_s, \psi_n, M)$ and $(z, \hat{z}) = (y, y_r)$, and the definition of Δ_μ in (10), we have

$$\begin{aligned}\frac{M}{2}\|y - y_r\|^2 &\leq (\widetilde{\psi}_s + \psi_n)(y) - (\widetilde{\psi}_s + \psi_n)(y_r)\\ &= \psi(y) - \psi(y_r) - \langle v, y - y_r\rangle \leq \Delta_\mu(y_r; y, v).\end{aligned}$$

(c) Using the assumption that $\Delta_\mu(y_r; y, v) \leq \varepsilon$, part (b), and the inequality in (9), we have that

$$\|y - y_r\| \leq \sqrt{\frac{2\Delta_\mu(y_r; y, v)}{M}} \leq \sqrt{\frac{2\varepsilon}{M}} \leq \frac{\theta}{\sqrt{M}}\|y - z_0\|. \tag{14}$$

Using the triangle inequality, the definition of $L(\cdot, \cdot)$, (14) and the inequality in (9) again, we conclude that

$$\|v_r\| \leq \|v\| + [M + L(y, y_r)] \cdot \|y - y_r\| \leq \theta\left[1 + \frac{M + L(y, y_r)}{\sqrt{M}}\right]\|y - z_0\|.$$

(d) The fact that $\Delta_\mu(u; y, v) \leq \varepsilon$ for every $u \in \operatorname{dom}\psi_n$ follows immediately from the inclusion in (9) and the definition of Δ_μ in (10). The fact that (13) holds now follows from part (c). □

We make a few remarks about Proposition 1. First, it follows from (a) that (y_r, v_r) satisfies the inclusion in (8). Second, it follows from (a) and (c) that if $\theta = 0$, then $(\varepsilon, v_r) = (0, 0)$, and hence y_r is an exact stationary point of (7). In general, (13) implies that the residual $\|v_r\|$ is directly proportional to $\|y - w\|$, and hence, becomes smaller as this quantity approaches *zero*.

Inequality (13) plays an important technical role in the complexity analysis of the two prox-type methods of Section 3. Sufficient conditions for its validity are provided in (c) and (d), with (c) being the weaker one, in view of (d). When ψ_s is convex, it is shown that every iterate of the R-ACG algorithm presented below always satisfies the inclusion in (9), and hence, verifying the the validity of the sufficient condition in (c) amounts to simply checking whether the inequality in (9) holds. When ψ_s is not convex, verification of the inclusion in (9), and hence the sufficient condition in (d), is generally not possible, while the one in (c) is. This is a major advantage of the sufficient condition in (c), which is exploited in this paper towards the development of adaptive prox-type methods which attempt to approximately solve (7) when ψ_s is not convex.

For the sake of future reference, we now state the following problem for finding a triple (y, v, ε) satisfying the sufficient condition in Proposition 1(c). Its statement relies on the refinement procedure preceding Proposition 1.

Problem $\mathcal{B}$: Given the same inputs as in Problem $\mathcal{A}$, find $(y, v, \varepsilon) \in \operatorname{dom}\psi \times \mathcal{Z} \times \mathbb{R}_+$ satisfying the inequality in (9) and

$$\Delta_\mu(y_r; y, v) \leq \varepsilon, \tag{15}$$

where $\Delta_\mu(\cdot;\cdot,\cdot)$ is as in (10) and y_r is the first component of the refined pair $(y_r, v_r) = RP(y, v)$.

We now state the aforementioned R-ACG algorithm which solves Problem $\mathcal{A}$ when ψ_s is convex and solves Problem $\mathcal{B}$ whenever ψ_s is not convex and two key inequalities are satisfied, one at every iteration (i.e., (16)) and one at the end of its execution.

R-ACG Algorithm

Input: a quadruple (μ, M, ψ_s, ψ_n) satisfying (B1)–(B2) and a pair (θ, z_0);
Output: a triple (y, v, ε) that solves Problem $\mathcal{B}$ or a *failure* status;

0. define $\psi := \psi_s + \psi_n$ and set $z_0^c = z_0$, $B_0 = 0$, $\Gamma_0 \equiv 0$, and $j = 1$;

1. compute the iterates

$$\begin{aligned}
\xi_{j-1} &= \frac{1 + \mu B_{j-1}}{M - \mu}, \quad b_{j-1} = \frac{\xi_{j-1} + \sqrt{\xi_{j-1}^2 + 4\xi_{j-1}B_{j-1}}}{2}, \\
B_j &= B_{j-1} + b_{j-1}, \quad \tilde{z}_{j-1} = \frac{B_{j-1}}{B_j} z_{j-1} + \frac{b_{j-1}}{B_j} z_{j-1}^c, \\
z_j &= \operatorname*{argmin}_{u \in \mathcal{Z}} \left\{ l_{\psi_s}(u; \tilde{z}_{j-1}) + \psi_n(u) + \frac{M}{2}\|u - \tilde{z}_{j-1}\|^2 \right\}, \\
z_j^c &= \frac{1}{1 + \mu B_j} \left[z_{j-1}^c - b_{j-1}(M - \mu)(\tilde{z}_{j-1} - z_j) + \mu(B_{j-1} z_{j-1}^c + b_{j-1} z_j) \right];
\end{aligned}$$

2. compute the quantities

$$\begin{aligned}
\tilde{\gamma}_j &= l_{\psi_s}(\cdot; \tilde{z}_{j-1}) + \psi_n + \frac{\mu}{2}\|\cdot - \tilde{z}_{j-1}\|^2, \\
\gamma_j &= \tilde{\gamma}_j(z_j) + (M - \mu)\langle \tilde{z}_{j-1} - z_j, \cdot - z_j \rangle + \frac{\mu}{2}\|\cdot - z_j\|^2, \\
\Gamma_j &= \frac{B_{j-1}}{B_j}\Gamma_{j-1} + \frac{b_{j-1}}{B_j}\gamma_j, \quad r_j = \frac{z_0^c - z_j^c}{B_j} + \mu(z_j^c - z_j), \\
\eta_j &= \max\left\{ 0, \psi(z_j) - \Gamma_j(z_j^c) - \left\langle r_j, z_j - z_j^c \right\rangle + \frac{\mu}{2}\|z_j - z_j^c\|^2 \right\};
\end{aligned}$$

3. if the inequality

$$\left(\frac{1}{1 + \mu B_j} \right) \|B_j r_j + z_j - z_0\|^2 + 2B_j\eta_j \leq \|z_j - z_0\|^2 \tag{16}$$

holds, then go to step 4; otherwise, **stop** with a *failure* status;

4. if the inequality

$$\|r_j\|^2 + 2\eta_j \leq \theta^2 \|z_j - z_0\|^2, \tag{17}$$

holds, then go to step 5; otherwise, go to step 1;

5. set $(y, v, \varepsilon) = (z_j, r_j, \eta_j)$ and compute $(y_r, v_r) = RP(z_j, r_j)$; if the condition

$$\Delta_\mu(y_r; y, v) \leq \varepsilon,$$

holds then **stop** with a *success* status and **output** the triple (y, v, ε); otherwise, **stop** with a *failure* status.

It is well-known (see, for example, [8, Proposition 2.3]) that the scalar B_j updated in step 1 satisfies

$$B_j \geq \frac{1}{M} \max\left\{\frac{j^2}{4}, \left(1 + \sqrt{\frac{\mu}{4M}}\right)^{2(j-1)}\right\} \quad \forall j \geq 1. \tag{18}$$

The next result presents the key properties about the R-ACG algorithm.

Proposition 2. *The R-ACG algorithm has the following properties:*

(a) it stops with either failure or success in

$$\mathcal{O}\left(\left[1 + \sqrt{\frac{L}{\mu}}\right] \log_1^+ \left[LK_\theta(1 + \mu K_\theta)\right]\right) \tag{19}$$

iterations, where $K_\theta := 1 + \sqrt{2}/\theta$*;*

(b) if it stops with success, then its output (y, v, ε) *solves Problem* $\mathcal{B}$*;*

(c) if ψ_s *is* μ*–strongly convex then it always stops with success and its output* (y, v, ε) *solves Problem* $\mathcal{A}$*.*

Proof. (a) See Appendix B.

(b) This follows from the successful checks in step 4 and 5 of the algorithm.

(c) The fact that the algorithm never stops with failure follows from Proposition 20(c)–(d) in Appendix B. The fact that the algorithm stops with success follows from the previous statement, the successful checks in step 4 and 5 of the algorithm, and the fact that the algorithm stops in a finite number of iterations in part (a). □

3 Inexact Composite Gradient Methods

This section presents the ICG methods and the general problem that they solve. It contains three subsections. The first one presents the problem of interest and gives a general outline of the ICG methods, the second one presents the IA-ICG method, and the third one presents the DA-ICG method. For the ease of presentation, the proofs in this section are deferred to Section 6.

3.1 Problem of Interest and Outline of the Methods

This subsection describes the problem that the ICG methods solve and outlines their structure.

Instead of considering problems having the spectral structure mentioned in Section 1, this section considers a more general NCO problem where its variable u lies in a finite dimensional inner product space $\mathcal{Z}$ (and, hence, can be either a vector and/or matrix) and presents both ICG methods in this more general setting. Section 4 then presents a modification of the ACG subroutine

used by both ICG methods that drastically improves their efficiency in the setting of the spectral problem (1).

More specifically, this section considers the problem

$$\min_{u\in\mathcal{Z}} [\phi(u) := f_1(u) + f_2(u) + h(u)] \tag{20}$$

where the functions f_1, f_2, and h are assumed to satisfy the following assumptions:

(A1) $h \in \overline{\text{Conv}}\ \mathcal{Z}$;

(A2) f_1, f_2 are continuously differentiable functions and there exists $(m_1, M_1) \in \mathbb{R}^2$ and $(m_2, M_2) \in \mathbb{R}^2$ such that, for $i \in \{1, 2\}$, we have

$$-\frac{m_i}{2}\|u - y\|^2 \leq f_i(u) - \ell_{f_i}(u; y) \leq \frac{M_i}{2}\|u - y\|^2 \quad \forall u, y \in \operatorname{dom} h; \tag{21}$$

(A3) for $i \in \{1, 2\}$, we have

$$\|\nabla f_i(u) - \nabla f_i(y)\| \leq L_i\|u - y\| \quad \forall u, y \in \operatorname{dom} h,$$

where $L_i := \max\{|m_i|, |M_i|\}$;

(A4) $\phi_* := \inf_{u\in\mathcal{Z}} \phi(u) > -\infty$.

Note that assumption (A2) implies that assumption (A3) holds when the interior of $\operatorname{dom} h$ is nonempty. Under the above assumptions, the proposed ICG methods find an approximate solution $(\hat{y}, \hat{v})$ of (20) as in (8) with $\psi_s = f_1 + f_2$ and $\psi_n = h$, i.e.

$$\hat{v} \in \nabla f_1(\hat{y}) + \nabla f_2(\hat{y}) + \partial h(\hat{y}), \quad \|\hat{v}\| \leq \hat{\rho}. \tag{22}$$

We now outline the ICG methods. Given a starting point $y_0 \in \operatorname{dom} \psi_n$ and a special stepsize $\lambda > 0$, each method continually calls the R-ACG algorithm of Section 2 to find an approximate solution of a prox-linear form of (20). More specifically, each R-ACG call is used to tentatively find an approximate solution of

$$\min_{u\in\mathcal{Z}} \left[\psi(u) = \lambda\left[\ell_{f_1}(u; z_0) + f_2(u) + h(u)\right] + \frac{1}{2}\|u - z_0\|^2\right], \tag{23}$$

for some reference point z_0. For the IA-ICG method, the point z_0 is y_0 for the first R-ACG call and is the last obtained approximate solution for the other R-ACG calls. For the DA-ICG method, the point z_0 is chosen in an accelerated manner.

From the output of the k^{th} R-ACG call, a refined pair $(\hat{y}, \hat{v}) = (\hat{y}_k, \hat{v}_k)$ is generated which: (i) always satisfies the inclusion of (22); and (ii) is such that $\min_{i\leq k} \|\hat{v}_i\| \to 0$ as $k \to \infty$. More specifically, this refined pair is generated by applying the refinement procedure of Section 2 and adding some adjustments to the resulting output to conform with our goal of finding an approximate solution as in (22). For the ease of future reference, we now state this specialized refinement procedure. Before proceeding, we introduce the shorthand notation

$$M_i^+ := \max\{M_i, 0\}, \quad m_i^+ := \max\{m_i, 0\}, \quad L_i(x, y) := L_{f_i}(x, y), \tag{24}$$

for $i \in \{1, 2\}$, to keep its presentation (and future results) concise.

Specialized Refinement Procedure

Input: a quadruple (M_2, f_1, f_2, h) satisfying (A1)–(A2), a scalar $\lambda > 0$, and a triple $(y, v, z_0) \in \operatorname{dom} \psi_n \times \mathcal{Z} \times \mathcal{Z}$;
Output: a pair $(\hat{y}, \hat{v})$ satisfying the inclusion of (22);

1. compute $(\hat{y}, v_r) = RP(y, v)$ using the refinement procedure in Section 2 with

$$M = \lambda M_2^+ + 1, \quad \psi_s = \lambda\,[\ell_{f_1}(\cdot; z_0) + f_2] + \frac{1}{2}\|\cdot - z_0\|^2, \quad \psi_n = \lambda h; \tag{25}$$

2. compute the residual

$$\hat{v} = \frac{1}{\lambda}(v_r + z_0 - y) + \nabla f_1(\hat{y}) - \nabla f_1(z_0),$$

and output $(\hat{y}, \hat{v})$.

The result below states some properties about the above procedure. For the sake of brevity, we write $(\hat{y}, \hat{v}) = SRP(y, v, z_0)$ to indicate that the pair $(\hat{y}, \hat{v})$ is the output of the above procedure with inputs (M_2, f_1, f_2, h), λ, and (y, v, z_0).

Lemma 3. *Let (m_1, M_1), (m_2, M_2), and (f_1, f_2, h) satisfying assumptions (A1)–(A3) and a quadruple $(z_0, y, v, \varepsilon) \in \mathcal{Z} \times \operatorname{dom}\psi_n \times \mathcal{Z} \times \mathbb{R}_+$ be given. Moreover, let $(\hat{y}, \hat{v}) = SRP(y, v, z_0)$ and define*

$$C_\lambda(x, y) := \frac{1 + \lambda\left[M_2^+ + L_1(x, y) + L_2(x, y)\right]}{\sqrt{1 + \lambda M_2^+}}, \tag{26}$$

for every $x, y \in \mathcal{Z}$. Then, the following statements hold:

(a) $\hat{v} \in \nabla f_1(\hat{y}) + \nabla f_2(\hat{y}) + \partial h(\hat{y})$;

(b) if (y, v, ε) solves Problem $\mathcal{B}$ with (μ, ψ_s, ψ_n) as in (28), then

$$\|\hat{v}\| \le \left[L_1(y, w) + \frac{2 + \theta C_\lambda(y, \hat{y})}{\lambda}\right]\|y - z_0\|.$$

It is worth recalling from Section 1 that in the applications we consider, the cost of the R-ACG call is small compared to SVD computation that is performed before solving each subproblem as in (23). Hence, in the analysis that follows, we present complexity results related to the number of subproblems solved rather than the total number of R-ACG iterations. We do note, however, that the number of R-ACG iterations per subproblem is finite in view of Proposition 2(a).

3.2 Static and Dynamic IA-ICG Methods

This subsection presents the static and dynamic IA-ICG methods.

We first state the static IA-ICG method.

Static IA-ICG Method

Input: function triple (f_1, f_2, h) and scalar quadruple $(m_1, M_1, m_2, M_2) \in \mathbb{R}^4$ satisfying (A1)–(A4), tolerance $\hat{\rho} > 0$, initial point $y_0 \in \operatorname{dom} h$, and scalar pair $(\lambda, \theta) \in \mathbb{R}_{++} \times (0, 1)$ satisfying

$$\lambda M_1 + \theta^2 \le \frac{1}{2}; \tag{27}$$

Output: a pair $(\hat{y}, \hat{v})$ satisfying (22) or a *failure* status;

0. let $\Delta_1(\cdot; \cdot, \cdot)$ be as in (10) with $\mu = 1$, and set $k = 1$;

1. use the R-ACG algorithm to tentatively solve Problem $\mathcal{B}$ associated with (23), i.e., with inputs (μ, M, ψ_s, ψ_n) and (θ, z_0) where the former is given by

$$\begin{aligned} \mu = 1, \quad M = \lambda M_2^+ + 1, \\ \psi_s = \lambda\left[\ell_{f_1}(\cdot; z_0) + f_2\right] + \frac{1}{2}\|\cdot - z_0\|^2, \quad \psi_n = \lambda h, \end{aligned} \tag{28}$$

   and $z_0 = y_{k-1}$; if the R-ACG stops with *failure*, then **stop** with a *failure* status; otherwise, let $(y_k, v_k, \varepsilon_k)$ denote its output and go to step 2;

2. if the inequality $\Delta_1(y_{k-1}; y_k, v_k) \le \varepsilon_k$ holds, then go to step 3; otherwise, **stop** with a *failure* status;

3. set $(\hat{y}_k, \hat{v}_k) = SRP(y_k, v_k, y_{k-1})$; if $\|\hat{v}_k\| \le \hat{\rho}$ then **stop** with a *success* status and **output** $(\hat{y}, \hat{v}) = (\hat{y}_k, \hat{v}_k)$; otherwise, update $k \leftarrow k + 1$ and go to step 1.

Note that the static IA-ICG method may fail without obtaining a pair satisfying (22). In Theorem 4(c) below, we state that a sufficient condition for the method to stop successfully is that f_2 be convex. This property will be important when we present the dynamic IA-ICG method, which: (i) repeatedly calls the static method; and (ii) incrementally transfers convexity from f_1 to f_2 between each call until a successful termination is achieved.

We now make some additional remarks about the above method. First, it performs two kinds of iterations, namely, ones that are indexed by k and ones that are performed by the R-ACG algorithm. We refer to the former kind as outer iterations and the latter kind as inner iterations. Second, in view of (27), if $M_1 > 0$ then $0 < \lambda < (1-2\theta^2)/(2M_1)$ whereas if $M_1 \le 0$ then $0 < \lambda < \infty$. Finally, the most expensive part of the method is the R-ACG call in step 1. In Section 4, we show that this call can be replaced with a call to a spectral version of R-ACG that is dramatically more efficient when the underlying problem has the spectral structure as in (1).

The next result summarizes some facts about the static IA-ICG method. Before proceeding, we first define some useful quantities. For $\lambda > 0$ and $u, w \in \mathcal{Z}$, define

$$\widetilde{\ell}_\phi(u; w) := \ell_{f_1}(u; w) + f_2(u) + h(u), \quad \overline{C}_\lambda := \frac{1 + \lambda(M_2^+ + L_1 + L_2)}{\sqrt{1 + \lambda M_2^+}}. \tag{29}$$

Theorem 4. *The following statements hold about the static IA-ICG method:*

(a) it stops in

$$\mathcal{O}_1\left(\left[\sqrt{\lambda}L_1 + \frac{1 + \theta\overline{C}_\lambda}{\sqrt{\lambda}}\right]^2 \left[\frac{\phi(z_0) - \phi_*}{\hat{\rho}^2}\right]\right) \tag{30}$$

outer iterations, where ϕ_* is as in (A4);

(b) if it stops with success, then its output pair $(\hat{y}, \hat{v})$ is a $\hat{\rho}$–approximate solution of (20);

(c) if f_2 is convex, then it always stops with success.

We now make three remarks about the above results. First, if $\theta = \mathcal{O}(1/\overline{C}_\lambda)$ then (30) is on the order of

$$\mathcal{O}_1\left(\left[\sqrt{\lambda}L_1 + \frac{1}{\sqrt{\lambda}}\right]^2 \left[\frac{\phi(z_0) - \phi_*}{\hat{\rho}^2}\right]\right). \tag{31}$$

Moreover, comparing the above complexity to the iteration complexity of the ECG method described in Section 1, which is known (see, for example, [14]) to obtain an approximate solution of (20) in

$$\mathcal{O}_1\left(\left[\sqrt{\lambda}(L_1+L_2)+\frac{1}{\sqrt{\lambda}}\right]^2\left[\frac{\phi(z_0)-\phi_*}{\hat{\rho}^2}\right]\right) \tag{32}$$

iterations, we see that (31) is smaller than (32) in magnitude when L_2 is large. Notice also that the complexity in (31) corresponds to applying the ECG method to (1) where the composite function is f_2+h instead of just h. Second, Theorem 4(b) shows that if the method stops with success, regardless of the convexity of f_2, then its output pair $(\hat{y},\hat{v})$ is always an approximate solution of (20). Third, in view of Proposition 10, the quantities L_1 and $\overline{C}_\lambda$ in all of the previous complexity results can be replaced by their averaged counterparts in (48). As these averaged quantities only depend on $\{(y_i,\hat{y}_i)\}_{i=1}^k$, we can infer that the static IA-ICG method adapts to the local geometry of its input functions.

We now state the dynamic IA-ICG method that resolves the issue of failure in the static IA-ICG method.

Dynamic IA-ICG Method

Input: the same as the static IA-ICG method but with an additional parameter $\xi_0>0$;
Output: a pair $(\hat{y},\hat{v})$ satisfying (22);

0. set $\xi=\xi_0$, $\ell=1$, and

$$\begin{gathered} f_1=f_1-\frac{\xi}{2}\|\cdot\|^2,\quad f_2=f_2+\frac{\xi}{2}\|\cdot\|^2,\\ m_1=m_1+\xi,\quad M_1=M_1-\xi,\quad m_2=m_2-\xi,\quad M_2=M_2+\xi; \end{gathered} \tag{33}$$

1. call the static IA-ICG method with inputs (f_1,f_2,h), (m_1,M_1,m_2,M_2), $\hat{\rho}$, y_0, and (λ,θ);
2. if the static IA-ICG call stops with a *failure* status, then set $\xi=2\xi$, update the quantities in (33) with the new value of ξ, increment $\ell=\ell+1$, and go to step 1; otherwise, let $(\hat{y},\hat{v})$ be the output pair returned by the static IA-ICG call, **stop**, and **output** this pair.

Some remarks about the above method are in order. First, in view of (27) and the fact that M_1 is monotonically decreasing, the parameter λ does not need to be changed for each IA-ICG call. Second, in view of assumption (A2) and Theorem 4(c), the IA-ICG call in step 1 always terminates with success whenever $m_2\leq 0$. As a consequence, the total number of IA-ICG calls is at most $\lceil\log(2m_2^+/\xi_0)\rceil$. Third, in view of the second remark and Theorem 4(b), the method always obtains a $\hat{\rho}$–approximate solution of (20) in a finite number of IA-ICG outer iterations. Finally, in view of second remark again, the total number of IA-ICG outer iterations is as in Theorem 4(a) but with: (i) an additional multiplicative factor of $\lceil\log(2m_2^+/\xi_0)\rceil$; and (ii) the constants m_1 and M_2 replaced with $(m_1+2m_2^+)$ and $(M_2+2m_2^+)$, respectively. It is worth mentioning that a more refined analysis, such as the one in [10], can be applied in order to remove the factor of $\lceil\log(2m_2^+/\xi_0)\rceil$ from the previously mentioned complexity.

3.3 Static and Dynamic DA-ICG Methods

This subsection presents the static DA-ICG method, but omits the statement of its dynamic variant for the sake of brevity. We do argue, however, that the dynamic variant can be stated in the same

way as the dynamic IA-ICG method of Subsection 6.1 but with the call to the static IA-ICG method replaced with a call to the static DA-ICG method of this subsection.

We start by stating some additional assumptions. It is assumed that:

(i) the set $\operatorname{dom} h$ is closed;

(ii) there exists a bounded set $\Omega \supseteq \operatorname{dom} h$ for which a projection oracle exists.

We now state the static DA-ICG method.

Static DA-ICG Method

Input: function triple (f_1, f_2, h) and scalar quadruple $(m_1, M_1, m_2, M_2) \in \mathbb{R}^4$ satisfying (A1)–(A4), tolerance $\hat{\rho} > 0$, initial point $y_0 \in \operatorname{dom} h$, and scalar pair $(\lambda, \theta) \in \mathbb{R}_{++} \times (0, 1)$ satisfying

$$\lambda M_1 + \theta^2 \leq \frac{1}{2}; \tag{34}$$

Output: a pair $(\hat{y}, \hat{v})$ satisfying (22) or a *failure* status;

0. let $\Delta_1(\cdot;\cdot,\cdot)$ be as in (10) with $\mu = 1$, and set $A_0 = 0$, $x_0 = y_0$, and $k = 1$;

1. compute the quantities

$$\begin{aligned} a_{k-1} &= \frac{1 + \sqrt{1 + 4A_{k-1}}}{2}, \quad A_k = A_{k-1} + a_{k-1}, \\ \tilde{x}_{k-1} &= \frac{A_{k-1} y_{k-1} + a_{k-1} x_{k-1}}{A_k}; \end{aligned} \tag{35}$$

2. use the R-ACG algorithm to tentatively solve Problem $\mathcal{B}$ associated with (23), i.e., with inputs (μ, M, ψ_s, ψ_n) and (θ, z_0) where the former is as in (28) and $z_0 = \tilde{x}_{k-1}$; if the R-ACG stops with *success*, then let $(y_k^a, v_k, \varepsilon_k)$ denote its output and go to step 3; otherwise, **stop** with a *failure* status;

3. if the inequality $\Delta_1(y_{k-1}; y_k^a, v_k) \leq \varepsilon_k$ holds, then go to step 4; otherwise, **stop** with a *failure* status;

4. set $(\hat{y}_k, \hat{v}_k) = SRP(y_k^a, v_k, \tilde{x}_{k-1})$ where $SRP(\cdot,\cdot,\cdot)$ is described in Subsection 3.1; if $\|\hat{v}_k\| \leq \hat{\rho}$ then **stop** with a *success* status and **output** $(\hat{y}, \hat{v}) = (\hat{y}_k, \hat{v}_k)$; otherwise, compute

$$\begin{aligned} x_k &= \operatorname*{argmin}_{u \in \Omega} \frac{1}{2} \left\| u - \left[x_{k-1} - a_{k-1} \left(v_k + \tilde{x}_{k-1} - y_k^a \right) \right] \right\|^2, \\ y_k &= \operatorname*{argmin}_{u \in \{y_{k-1}, y_k^a\}} \left[f_1(u) + f_2(u) + h(u) \right], \end{aligned} \tag{36}$$

   update $k \leftarrow k + 1$, and go to step 1.

Note that, similar to the static IA-ICG method, the static DA-ICG method may fail without obtaining a pair satisfying (22). Proposition 5(c) shows that a sufficient condition for the method to stop successfully is that f_2 be convex. Using arguments similar to the ones employed to derive the dynamic IA-ICG method, a dynamic version of DA-ICG method can also be developed that repeatedly invokes the static DA-ICG in place of the static IA-ICG.

We now make some additional remarks about the above method. First, it performs two kinds of iterations, namely, ones that are indexed by k and ones that are performed by the R-ACG algorithm. We refer to the former kind as outer iterations and the latter kind as inner iterations. Second, in view of the update for y_k in (36), the collection of function values $\{\phi(y_i)\}_{i=0}^k$ is non-increasing. Third, in view of (34), if $M_1 > 0$ then $0 < \lambda < (1-2\theta^2)/(2M_1)$ whereas if $M_1 \leq 0$ then $0 < \lambda < \infty$. Finally, the most expensive part of the method is the R-ACG call in step 2. In Section 4, we show that this call can be replaced with a call to a spectral version of R-ACG that is dramatically more efficient when the underlying problem has the spectral structure as in (1).

It is worth mentioning that the outer iteration scheme of the DA-ICG method is a monotone and inexact generalization of the A-ECG method in [6]. More specifically, this A-ECG method is a version of the DA-ICG method where: (i) $\theta = 0$; (ii) the R-ACG algorithm in step 2 is replaced by an exact solver of (23); (iii) the update of x_k in (36) is replaced by an update involving the prox evaluation of the function $a_{k-1}h$; and (iv) both f_1 and f_2 are linearized instead of just f_2 in the DA-ICG method. Hence, the DA-ICG method can be significantly more efficient when its R-ACG call is more efficient than an exact solver of (23) and/or when the projection onto Ω is more efficient than evaluating the prox of $a_{k-1}h$.

The next result summarizes some facts about the DA-ICG method. Before proceeding, we introduce the useful constants

$$\begin{aligned} D_h &:= \sup_{u,z\in\operatorname{dom} h} \|u-z\|, \quad D_\Omega := \sup_{u,z\in\Omega} \|u-z\|, \quad \Delta_\phi^0 := \phi(y_0) - \phi_*, \\ d_0 &:= \inf_{u^*\in\mathcal{Z}} \{\|y_0 - u^*\| : \phi(u^*) = \phi_*\}, \quad E_{\lambda,\theta} := \sqrt{\lambda} L_1 + \frac{1+\theta\overline{C}_\lambda}{\sqrt{\lambda}}. \end{aligned} \tag{37}$$

Theorem 5. *The following statements hold about the static DA-ICG method:*

(a) it stops in

$$\mathcal{O}_1\left(\frac{E_{\lambda,\theta}^2[m_1^+ D_h^2 + \Delta_\phi^0]}{\hat{\rho}^2} + \frac{E_{\lambda,\theta}[m_1^+ + 1/\lambda]^{1/2} D_\Omega}{\hat{\rho}}\right) \tag{38}$$

outer iterations;

(b) if it stops with success, then its output pair $(\hat{y}, \hat{v})$ is a $\hat{\rho}$–approximate solution of (20);

(c) if f_2 is convex, then it always stops with success in

$$\mathcal{O}_1\left(\frac{E_{\lambda,\theta}^2 m_1^+ D_h^2}{\hat{\rho}^2} + \frac{E_{\lambda,\theta}[m_1^+]^{1/2} D_\Omega}{\hat{\rho}} + \frac{E_{\lambda,\theta}^{2/3} d_0^{2/3} \lambda^{-1/3}}{\hat{\rho}^{2/3}}\right) \tag{39}$$

outer iterations.

We now make three remarks about the above results. First, in the "best" scenario of $\max\{m_1, m_2\} \leq 0$, i.e., f_1 and f_2 are convex, we have that (39) reduces to

$$\mathcal{O}_1\left(\left[L_1 + \frac{1}{\lambda}\right]^{2/3}\left[\frac{d_0^{2/3}}{\hat{\rho}^{2/3}}\right]\right),$$

which has a smaller dependence on $\hat{\rho}$ when compared to (31). In the "worst" scenario of $\min\{m_1, m_2\} > 0$, if we take $\theta = \mathcal{O}(1/\overline{C}_\lambda)$, then (38) reduces to

$$\mathcal{O}_1\left(\left[\sqrt{\lambda} L_1 + \frac{1}{\sqrt{\lambda}}\right]^2\left[\frac{m_1^+ D_h^2 + \phi(y_0) - \phi_*}{\hat{\rho}^2}\right]\right),$$

which has the same dependence on $\hat{\rho}$ as in (31). Second, part (c) shows that if the method stops with an output pair $(\hat{y}, \hat{v})$, regardless of the convexity of f_2, then that pair is always an approximate solution of (20). Third, in view of Proposition 18, the quantities L_1 and $\overline{C}_\lambda$ in all of the previous complexity results can be replaced by their averaged counterparts in (63). As these averaged quantities only depend on $\{(y_i^a, \hat{y}_i, \tilde{x}_{i-1})\}_{i=1}^k$, we can infer that the static DA-ICG method, like the static IA-ICG method of the previous subsection, also adapts to the local geometry of its input functions.

4 Exploiting the Spectral Decomposition

Recall that at every outer iteration of the ICG methods in Section 3, a call to the R-ACG algorithm is made to tentatively solve Problem $\mathcal{B}$ (see Subsection 3.1) associated with (23). Our goal in this section is to present a more efficient version of R-ACG (based on the idea outlined in Section 1) when the underlying problem has the spectral structure as in (1).

The content of this section is divided into two subsections. The first one presents the aforementioned algorithm, whereas the second one proves its key properties.

4.1 Spectral R-ACG Algorithm

This subsection presents the R-ACG algorithm mentioned above. Throughout our presentation, we let Z_0 represent the starting point given to the R-ACG algorithm by the two ICG methods.

We first state the aforementioned efficient algorithm.

Spectral R-ACG Algorithm

Input: a quadruple $(M_2, f_1, f_2^{\mathcal{V}}, h^{\mathcal{V}})$ satisfying (A1)–(A3) with $(f_2, h) = (f_2^{\mathcal{V}}, h^{\mathcal{V}})$ and a triple (λ, θ, Z_0);
Output: a triple (Y, V, ε) that solves Problem $\mathcal{B}$ associated with (23) or a *failure* status;

1. compute
$$Z_0^\lambda := Z_0 - \lambda \nabla f_1(Z_0), \tag{40}$$
and a pair $(P, Q) \in \mathcal{U}^m \times \mathcal{U}^n$ satisfying $Z_0^\lambda = P[\operatorname{dg} \sigma(Z_0^\lambda)]Q^*$;

2. use the R-ACG algorithm to tentatively solve Problem $\mathcal{B}$ associated with (4), i.e., with inputs $(\mu, M, \psi_s^{\mathcal{V}}, \psi_n^{\mathcal{V}})$ and (θ, z_0) where the former is given by
$$\begin{aligned} \mu = 1, \quad M := \lambda M_2^+ + 1, \\ \psi_s^{\mathcal{V}} := \lambda f_2^{\mathcal{V}} - \langle \sigma(Z_0^\lambda), \cdot \rangle + \frac{1}{2}\|\cdot\|^2, \quad \psi_n^{\mathcal{V}} := \lambda h^{\mathcal{V}}, \end{aligned} \tag{41}$$
and $z_0 = \operatorname{Dg}(P^* Z_0 Q)$; if the R-ACG stops with *success*, then let (y, v, ε) denote its output and go to step 3; otherwise, **stop** with a *failure* status;

3. set $Y = P(\operatorname{dg} y)Q^*$ and $V = P(\operatorname{dg} v)Q^*$, and output the triple (Y, V, ε).

We now make three remarks about the above algorithm. First, the matrices P and Q in step 1 can be obtained by computing an SVD of Z_0^λ. Second, in view of Proposition 20(a) and the fact that (μ, M) in (41) and (28) are the same, the iteration complexity is the same as the vanilla

R-ACG algorithm. Finally, because the functions $\psi_s^{\mathcal{V}}$ and $\psi_n^{\mathcal{V}}$ in (41) have vector inputs over $\mathbb{R}^r$, the steps in the spectral R-ACG algorithm are significantly less costly than the ones in the R-ACG algorithm, which use functions with matrix inputs over $\mathbb{R}^{m\times n}$.

The following result, whose proof is in the next subsection, presents the key properties of this algorithm.

Proposition 6. *The spectral R-ACG algorithm has the following properties:*

(a) if it stops with success, then its output triple (Y, V, ε) solves Problem $\mathcal{B}$ associated with (23);

(b) if f_2 is convex, then it always stops with success and its output (Y, V, ε) solves Problem $\mathcal{A}$ associated with (23).

4.2 Proof of Proposition 6

For the sake of brevity, let (ψ_s, ψ_n) be as in (28) and, using P and Q from the spectral R-ACG algorithm, define for every $(u, U) \in \mathbb{R}^r \times \mathbb{R}^{m\times n}$, the functions

$$\begin{aligned} \mathcal{M}(u) &:= P(\operatorname{dg} u)Q^*, \quad \mathcal{V}(U) := \operatorname{Dg}(P^*UQ), \\ \psi(U) &:= \psi_s(U) + \psi_n(U), \quad \psi^{\mathcal{V}}(u) := \psi_s^{\mathcal{V}}(u) + \psi_n^{\mathcal{V}}(u). \end{aligned}$$

The first result relates (ψ_s, ψ_n) to $(\psi_s^{\mathcal{V}}, \psi_n^{\mathcal{V}})$.

Lemma 7. *Let (y, v, ε) and (Y, V) be as in the spectral R-ACG algorithm. Then, the following properties hold:*

(a) we have

$$\psi_n^{\mathcal{V}}(y) = \psi_n(Y), \quad \psi_s^{\mathcal{V}}(y) + B_0^{\lambda} = \psi_s(Y),$$

where $B_0^{\lambda} := \lambda f_1(Z_0) - \lambda\langle \nabla f_1(Z_0), Z_0\rangle + \|Z_0\|_F^2/2$;

(b) we have

$$V \in \partial_{\varepsilon}\left(\psi - \frac{1}{2}\|\cdot - Y\|_F^2\right)(Y) \iff v \in \partial_{\varepsilon}\left(\psi^{\mathcal{V}} - \frac{1}{2}\|\cdot - y\|^2\right)(y). \tag{42}$$

Proof. (a) The relationship between $\psi_n^{\mathcal{V}}$ and ψ_n is immediate. On the other hand, using the definitions of Y, f_2, and B_0^{λ}, we have

$$\begin{aligned} \psi_s^{\mathcal{V}}(y) + B_0^{\lambda} &= \lambda f_2(Y) - \langle Z_0^{\lambda}, Y\rangle + \frac{1}{2}\|Y\|_F^2 + B_0^{\lambda} \\ &= \lambda\left[f_2(Y) + f_1(Z_0) + \langle \nabla f_1(Z_0), Y - Z_0\rangle\right] + \frac{1}{2}\|Y - Z_0\|_F^2 = \psi_s(Y). \end{aligned}$$

(b) Let $S_0 = V + Z_0^{\lambda} - Y$ and $s_0 = v + \sigma(Z_0^{\lambda}) - y$, and note that $S_0 = \mathcal{M}(s_0)$. Moreover, in view of part (a) and the definition of ψ, observe that the left inclusion in (42) is equivalent to $S_0 \in \partial_{\varepsilon}(\lambda[f_2 + h])(Y)$. Using this observation, the fact that S_0 and Y have a simultaneous SVD, and Theorem 23 with $(S, s) = (S_0, s_0)$, $\Psi = \lambda[f_2 + h]$, and $\Psi^{\mathcal{V}} = \lambda[f_2^{\mathcal{V}} + h^{\mathcal{V}}]$, we have that the left inclusion in (42) is also equivalent to $s_0 \in \partial_{\varepsilon}(\lambda[f_2^{\mathcal{V}} + h^{\mathcal{V}}])(y)$. The conclusion now follows from the observation that the latter inclusion is equivalent to the the right inclusion in (42). □

We are now ready to give the proof of Proposition 6.

Proof of Proposition 6. (a) Since $(y, v) = (\mathcal{V}(Y), \mathcal{V}(V))$, notice that the successful termination of the algorithm implies that the inequality in (9) and (15) hold. Using this remark, the fact that $\|V\|_F^2 = \|v\|^2$, and the bound

$$\begin{aligned}\theta^2\|z_j - z_0\|^2 &= \theta^2\left(\|z_j\|^2 - 2\langle z_j, \mathcal{V}(z_0)\rangle + \|Z_0\|_F^2\right) + \theta^2(\|\mathcal{V}(z_0)\|^2 - \|Z_0\|_F^2) \\ &\leq \theta^2\left(\|Z_j\|_F^2 - 2\langle Z_j, Z_0\rangle + \|Z_0\|_F^2\right) = \theta^2\|Z_j - Z_0\|_F^2, \end{aligned} \tag{43}$$

we then have that the inequality in (9) also holds with $(y, v) = (Y, V)$.

To show the corresponding inequality for (15), let $(Y_r, V_r) = RP(Y, V)$ using the refinement procedure in Section 2. Moreover, let $(y_r, v_r) = RP(y, v)$ and $\Delta_1^{\mathcal{V}}(\cdot;\cdot,\cdot)$ be as in (10), where $(\psi_s, \psi_n) = (\psi_s^{\mathcal{V}}, \psi_n^{\mathcal{V}})$. It now follows from (11), (12), Lemma 22 with $\Psi = \psi_n$ and $S = V + MY - \nabla\psi_s(Y)$, and Lemma 21(b) that Y_r, Y, V, and V_r have a simultaneous SVD. As a consequence of this, the first remark, and Lemma 7(a), we have that

$$\begin{aligned}\varepsilon \geq \Delta_1^{\mathcal{V}}(y_r; y, v) &= \psi^{\mathcal{V}}(y) - \psi^{\mathcal{V}}(y_r) - \langle v, y - y_r\rangle + \frac{1}{2}\|y_r - y\|^2 \\ &= \psi(Y) - \psi(Y_r) - \langle V, Y - Y_r\rangle + \frac{1}{2}\|Y_r - Y\|^2 = \Delta_1(Y_r; Y, V),\end{aligned}$$

and hence that (15) holds with $(y, v) = (Y, V)$.

(b) This follows from part (a), Proposition 2(c), and Lemma 7(b). □

5 Computational Results

This section presents computational results that highlight the performance of the dynamic IA-ICG and dynamic DA-ICG methods, and it contains three subsections. The first one describes the implementation details, the second presents computational results related to a set of spectral composite problem, while the third gives some general comments about the computational results.

5.1 Implementation Details

This subsection precisely describes the implementation of the methods and experiments of this section. Moreover, all of the code needed to replicate these experiments is readily available online[1].

We first describe some practical modifications to the dynamic IA-ICG method. Given $\lambda > 0$ and $(z_j, z_0) \in \mathcal{Z}^2$, denote

$$\Delta_\phi^\lambda = 4\lambda\left[\phi(z_0) - \widetilde{\ell}_\phi(z_j; z_0) - \frac{M_1}{2}\|z_j - z_0\|^2\right]$$

where $\widetilde{\ell}_\phi$ is as in (29). Motivated by the first inequality in the descent condition (46), we relax (17) in the R-ACG call to the three separate conditions: $\|z_j - z_0\|^2 \leq \Delta_\phi^\lambda$, $\|r_j\|^2 \leq \Delta_\phi^\lambda$, and $2\eta_j \leq \Delta_\phi^\lambda$.

We now describe some modifications and parameter choices that are common to both methods. First, both ICG methods use the spectral R-ACG algorithm of Subsection 4.1 in place of the R-ACG algorithm of Section 2. Moreover, this R-ACG variant uses a line search subroutine for estimating the upper curvature M that is used during its execution. Second, when each of the dynamic ICG methods invokes their static counterparts, the parameters A_0 and y_0 are set to be the last obtained parameters of the previous invocation or the original parameters if it is the first invocation, i.e., we

[1] See https://github.com/wwkong/nc_opt/tree/master/tests/papers/icg.

implement a warm–start strategy. Third, we adaptively update λ at each outer iteration as follows: given the old value of $\lambda = \lambda_{\text{old}}$ at the k^{th} outer iteration, the new value of $\lambda = \lambda_{\text{new}}$ at the $(k+1)^{\text{th}}$ iteration is given by

$$\lambda_{\text{new}} = \begin{cases} \lambda_{\text{old}}, & r_k \in [0.5, 2.0], \\ \lambda_{\text{old}} \cdot \sqrt{0.5}, & r_k < 0.5, \\ \lambda_{\text{old}} \cdot \sqrt{2}, & r_k > 2.0, \end{cases} \qquad r_k = \frac{\left[\lambda(M_2^+ + 2m_2^+) + 1\right] \|y_k - \hat{y}_k\|}{\|\hat{v}_k - \left[\lambda(M_2^+ + 2m_2^+) + 1\right](y_k - \hat{y}_k)\|}.$$

Fourth, we take $\mu = 1/2$ rather than $\mu = 1$ for each of R-ACG calls in order to reduce the possibility of a failure from the R-ACG algorithm. Fifth, in view of (43), we relax condition (17) in the vector-based R-ACG call of Subsection 4.1 to

$$\|r_j\|^2 + 2\eta_j \leq \theta^2 \|z_j - z_0\|^2 + \tau,$$

where $\tau := \theta^2(\|Z_0\|_F^2 - \|z_0\|^2) \geq 0$. Finally, both ICG methods choose the common hyperparameters $(\xi_0, \lambda, \theta) = (M_1, 5/M_1, 1/2)$ at initialization.

We now describe the five other benchmark methods considered. Throughout their descriptions, we let $m = m_1 + m_2$, $M = M_1 + M_2$, and $L = \max\{m, M\}$. The first method is Nesterov's efficient ECG method of [16] with $(\lambda, \gamma_u, \gamma_d) = (100/L, 2, 2)$. The second method is the accelerated inexact proximal point (AIPP) method of [10] with $(\lambda, \theta, \tau) = (1/m, 4, 10[\lambda M + 1])$ and the R-AIPPv2 stepsize scheme. The third method is a variant of the A-ECG method of [6, Algorithm 2], which we abbreviate as AG. In particular, this variant chooses its parameters as in [6, Corollary 2] with L_Ψ replaced by M, i.e., $\beta_k = 1/(2M)$ for every k (implying a more aggressive stepsize policy). It is worth mentioning that we tested the more conservative AG variant with $\beta_k = 1/(2L_\Psi)$ and observed that it performed substantially less efficient than the above aggressive variant. The fourth method is a special implementation of the adaptive A-ECG method in [7] with $(\gamma_1, \gamma_2, \gamma_3) = (0.4, 0.4, 1.0)$ and $(\delta, \sigma) = (10^{-2}, 10^{-10})$, which we abbreviate as UP. More specifically, we consider the UPFAG-fullBB method described in [7, Section 4], which uses a Barzilai-Borwein type stepsize selection strategy. The last is the A-ECG method of [13], named NC-FISTA, with $(\xi, \lambda) = (1.05m, 0.99/M)$, which we abbreviate as NCF.

Finally, we state some additional details about the numerical experiments. First, the problems considered are of the form in (1) and satisfy assumptions (A1)–(A4) with $f_2 = f_2^{\mathcal{V}} \circ \sigma$ and $h = h^{\mathcal{V}} \circ \sigma$. Second, given a tolerance $\hat{\rho} > 0$ and an initial point $Y_0 \in \operatorname{dom} h$, every method in this section seeks a pair $(\hat{Y}, \hat{V}) \in \operatorname{dom} h \times \mathbb{R}^{m \times n}$ satisfying

$$\hat{V} \in \nabla f_1(\hat{Y}) + \nabla(f_2^{\mathcal{V}} \circ \sigma)(\hat{Y}) + \partial(h^{\mathcal{V}} \circ \sigma)(\hat{Y}), \quad \frac{\|\hat{V}\|}{\|\nabla f_1(Y_0) + (f_2^{\mathcal{V}} \circ \sigma)(Y_0)\| + 1} \leq \hat{\rho},$$

and stops after 1000 seconds if such a point cannot be found. Third, to be concise, we abbreviate the IA-ICG and DA-ICG methods as IA and DA, respectively. Finally, all described algorithms are implemented in MATLAB 2020a and are run on Linux 64-bit machines that contain at least 8 GB of memory.

5.2 Spectral Composite Problems

This subsection presents computational results of a set of spectral composite optimization problems and contains two sub-subsections. The first one examines a class of nonconvex matrix completion problems, while the second one examines a class of blockwise matrix completion problems.

Name	ℓ	n	% nonzero	$\min_{i,j} A_{ij}$	$\max_{i,j} A_{ij}$
Anime[2]	506	9437	10.50%	1	10
FilmTrust[3]	1508	2071	1.14%	0.5	8

Table 1: Description of the MC data matrices $A \in \mathbb{R}^{m\times n}$.

5.2.1 Matrix completion

Given a quadruple $(\alpha, \beta, \mu, \theta) \in \mathbb{R}^4_{++}$, a data matrix $A \in \mathbb{R}^{\ell\times n}$, and indices Ω, this subsection considers the following constrained matrix completion (MC) problem:

$$\begin{aligned} \min_{U\in\mathbb{R}^{m\times n}} \ & \frac{1}{2}\|P_\Omega(U-A)\|_F^2 + \kappa_\mu \circ \sigma(U) + \tau_\alpha \circ \sigma(U) \\ \text{s.t.} \ & \|U\|_F^2 \le \sqrt{\ell n}\cdot \max_{i,j} |A_{ij}|, \end{aligned}$$

where P_Ω is the linear operator that zeros out any entry that is not in Ω and

$$\kappa_\mu(z) = \frac{\mu\beta}{\theta}\sum_{i=1}^n \log\left(1+\frac{|z_i|}{\theta}\right), \quad \tau_\alpha(z) = \alpha\beta\left[1-\exp\left(-\frac{\|z\|_2^2}{2\theta}\right)\right]$$

for every $z \in \mathbb{R}^n$. Here, the function $\kappa_\mu + \tau_\alpha$ is a nonconvex generalization of the convex elastic net regularizer (see, for example, [18]), and it is well-known (see, for example, [21]) that the function $\kappa_\mu - \mu\|\cdot\|_*$ is concave, differentiable, and has a $(2\beta\mu/\theta^2)$-Lipschitz continuous gradient.

We now describe the different data matrices that are considered. Each matrix $A \in \mathbb{R}^{\ell\times n}$ is obtained from a different collaborative filtering system where each row represents a unique user, each column represents a unique item, and each entry represents a particular rating. Table 1 lists the names of each data set, where the data originates from (in the footnotes), and some basic statistics about the matrices.

We now describe the experiment parameters considered. First the starting point Z_0 is randomly generated from a shifted binomial distribution that closely follows the data matrix A. More specifically, the entries of Z_0 are distributed according to a $\textsc{Binomial}(a, \mu/a) - \underline{A}$ distribution, where μ is the sample average of the nonzero entries in A, the integer a is the ceiling of the range of ratings in A, and $\underline{A}$ is the minimum rating in A. Second, the decomposition of the objective function is as follows

$$f_1 = \frac{1}{2}\|P_\Omega(\cdot - A)\|_F^2, \quad f_2^{\mathcal{V}} = \mu\left[\kappa_\mu(\cdot) - \frac{\beta}{\theta}\|\cdot\|_1\right] + \tau_\alpha(\cdot), \quad h^{\mathcal{V}} = \frac{\mu\beta}{\theta}\|\cdot\|_1 + \delta_{\mathcal{F}}(\cdot), \tag{44}$$

where $\mathcal{F} = \{U \in \mathbb{R}^{m\times n} : \|U\|_F \le \sqrt{\ell n}\cdot \max_{i,j}|A_{ij}|\}$ is the set of feasible solutions. Third, in view of the previous decomposition, the curvature parameters are set to be

$$m_1 = 0, \quad M_1 = 1, \quad m_2 = \frac{2\beta\mu}{\theta^2} + \frac{2\alpha\beta}{\theta}\exp\left(\frac{-3\theta}{2}\right), \quad M_2 = \frac{\alpha\beta}{\theta}, \tag{45}$$

where it can be shown that the smallest and largest eigenvalues of $\nabla^2\tau_\alpha(z)$ are bounded below and above by $-2\alpha\beta\exp(-3\theta/2)/\theta$ and $\alpha\beta/\theta$, respectively, for every $z \in \mathbb{R}^n$. Finally, each problem instance uses a specific data matrix A from Table 1, the hyperparameters $(\alpha, \beta, \mu) = (10, 20, 2)$ and

[2]See the subset of the ratings from https://www.kaggle.com/CooperUnion/anime-recommendations-database where each user has rated at least 720 items.

[3]See the ratings in the file "ratings.txt" under the FilmTrust section in https://www.librec.net/datasets.html.

$\hat{\rho} = 10^{-6}$, different values of the parameter θ, and Ω to be the index set of nonzero entries in the chosen matrix A.

We now present the results. Figure 1 presents two subplots for the results of the Anime dataset under a value of $\theta = 10^{-1}$. The first subplot contains the log objective value against runtime, while the second one contains the log of the minimal subgradients, i.e. $\min_{i \le k} \|\hat{V}_i\|$, against runtime. Tables 2 to 3 present the minimal subgradient size obtained within the time limit of 1000. Moreover, each row of these tables corresponds to a different choice of θ and the bolded numbers highlight which algorithm performed the best in terms of the size obtained in a run.

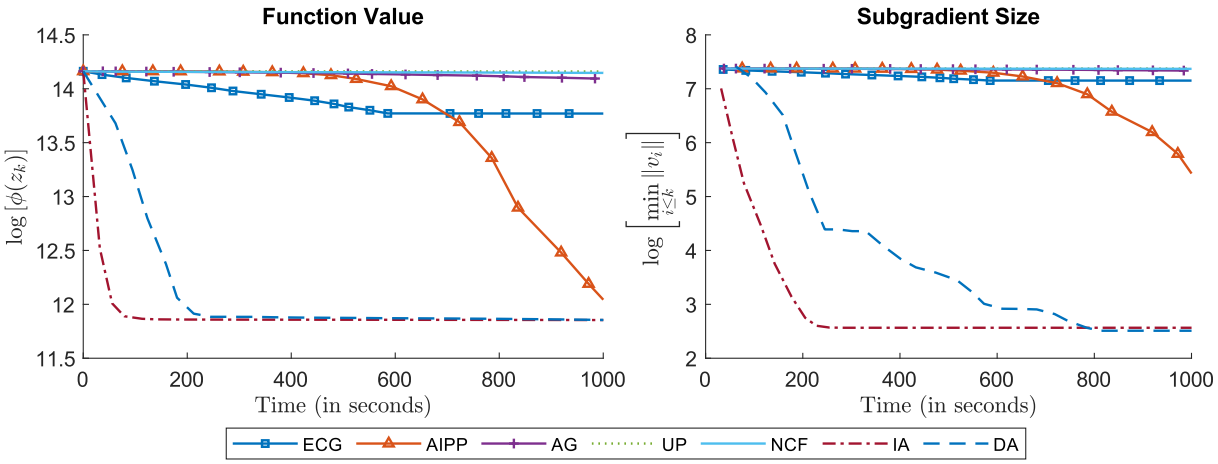

Figure 1: Function values and minimum subgradients for the Anime dataset with $\theta = 10^{-1}$.

Parameters $(\theta, m, M)^T$	Time t	Minimum Subgradient Size ($\min_{i \le k} \|\hat{V}_i\|$)						
		ECG	AIPP	AG	UP	NCF	IA	DA
$\begin{bmatrix} 1 \\ 169 \\ 201 \end{bmatrix}$	100	1088.6	1421.0	1568.9	1599.4	1488.8	**13.0**	78.5
	200	1088.6	221.9	1510.2	132.6	1362.4	**11.6**	39.2
	400	1088.6	55.6	1284.6	**7.5**	1147.9	11.6	11.1
	800	1088.6	7.7	716.7	**7.5**	862.7	11.6	11.1
$\begin{bmatrix} 0.1 \\ 11443 \\ 2001 \end{bmatrix}$	100	1542.0	1595.8	1593.8	-	1595.0	**189.7**	1345.1
	200	1489.9	1595.0	1591.5	1595.2	1594.2	**23.1**	378.1
	400	1391.0	1587.8	1584.3	1595.1	1592.3	**13.0**	60.6
	800	1276.3	990.5	1557.2	1594.3	1589.2	**13.0**	13.0
$\begin{bmatrix} 0.01 \\ 839400 \\ 20001 \end{bmatrix}$	100	1594.6	1595.9	1595.6	1595.8	1595.9	**162.9**	452.0
	200	1592.8	1595.6	1595.0	1595.8	1595.8	**33.5**	68.0
	400	1589.8	1569.5	1592.2	1595.8	1595.8	15.3	**14.7**
	800	1583.8	861.8	1582.3	1595.7	1595.7	15.3	**14.1**

Table 2: Minimum subgradient sizes for the Anime dataset. Times are in seconds and "-" indicates a run that did not generate a subgradient within the given time limit.

5.2.2 Blockwise matrix completion

Given a quadruple $(\alpha, \beta, \mu, \theta) \in \mathbb{R}^4_{++}$, a block decomposable data matrix $A \in \mathbb{R}^{\ell \times n}$ with blocks $\{A_i\}_{i=1}^k \subseteq \mathbb{R}^{p \times q}$, and indices Ω, this subsection considers the following constrained blockwise matrix

Parameters $(\theta, m, M)^T$	Time t	Minimum Subgradient Size ($\min_{i \le k} \|\hat{V}_i\|$)						
		ECG	AIPP	AG	UP	NCF	IA	DA
$\begin{bmatrix} 1 \\ 169 \\ 201 \end{bmatrix}$	100	127.1	328.6	328.5	-	326.2	**77.7**	342.4
	200	106.7	326.2	326.8	330.0	319.4	**60.7**	203.1
	400	106.7	294.6	319.2	330.0	305.9	**60.7**	186.4
	800	106.7	107.4	291.0	251.9	280.5	**60.7**	186.4
$\begin{bmatrix} 0.1 \\ 11443 \\ 2001 \end{bmatrix}$	100	309.0	330.0	329.6	329.9	329.9	**71.0**	242.3
	200	287.0	326.9	327.8	329.9	329.5	**71.0**	235.4
	400	248.0	188.7	321.9	329.8	328.8	**71.0**	202.7
	800	186.9	188.7	301.8	329.4	327.4	**71.0**	202.7
$\begin{bmatrix} 0.01 \\ 839400 \\ 20001 \end{bmatrix}$	100	330.1	330.2	330.2	-	330.2	**91.8**	263.9
	200	330.0	330.2	330.2	330.2	330.2	**91.8**	262.1
	400	329.7	330.2	330.1	330.2	330.2	**91.8**	262.1
	800	329.2	328.7	329.7	330.2	330.2	**91.8**	262.1

Table 3: Minimum subgradient sizes for the FilmTrust dataset. Times are in seconds and "-" indicates a run that did not generate a subgradient within the given time limit.

completion (BMC) problem:

$$\begin{aligned} \min_{U \in \mathbb{R}^{m \times n}} \; & \frac{1}{2}\|P_\Omega(U - A)\|_F^2 + \sum_{i=1}^{k} [\kappa_\mu \circ \sigma(U_i) + \tau_\alpha \circ \sigma(U_i)] \\ \text{s.t.} \; & \|U\|_F^2 \le \sqrt{\ell n} \cdot \max_{i,j} |A_{ij}|, \end{aligned}$$

where P_Ω, κ_μ, and τ_α are as in Subsection 5.2.1 and $U_i \in \mathbb{R}^{p \times q}$ is the i^{th} block of U with the same indices as A_i with respect to A.

We now describe the two classes of data matrices that are considered. Every data matrix is a 5-by-5 block matrix consisting of 50-by-100 sized submatrices. Every submatrix contains only 25% nonzero entries and each data matrix generates its submatrix entries from different probability distributions. More specifically, for a sampled probability $p \sim \textsc{Uniform}[0, 1]$ specific to a fixed submatrix, one class uses a $\textsc{Binomial}(n, p)$ distribution with $n = 10$, while the other uses a $\textsc{TruncatedNormal}(\mu, \sigma)$ distribution with $\mu = 10p$, $\sigma^2 = 10p(1 - p)$, and upper and lower bounds 0 and 10, respectively.

We now describe the experiment parameters considered. First, the the decomposition of the objective function and the quantities Z_0, (m_1, M_1), (m_2, M_2), $\hat{\rho}$, and Ω are the same as in Subsection 5.2.1. Second, we fix $(\alpha, \beta, \mu) = (10, 20, 2)$ and vary (θ, A) across the different problem instances.

We now present the results. Figure 2 contains the plots of the log objective function value against the runtime for the binomial data set, listed in increasing order of M_2. The corresponding plots for the truncated normal data set are similar to the binomial plots so we omit them for the sake of brevity. Tables 4 and 5 present the minimal subgradient size obtained within the time limit of 1000. Moreover, each row of these tables corresponds to a different choice of θ and the bolded numbers highlight which algorithm performed the best in terms of the size obtained in a run.

5.3 General Comments

This subsection makes two comments about the results obtained in the previous subsection. First, within the alloted time (i.e., 1000 seconds), the DA-ICG and IA-ICG methods obtained approximate

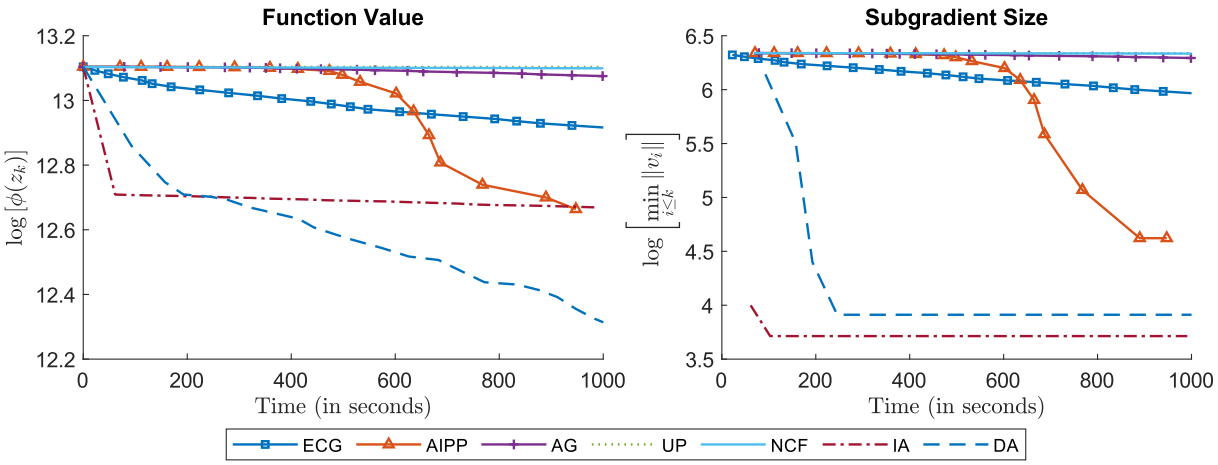

Figure 2: Function values and minimum subgradients for the truncated normal dataset with $\theta = 10^{-1}$.

solutions with small primal residual $\|\hat{V}_k\|$ much faster than the other first-order methods. More specifically, the former methods were able to obtain higher quality solutions much sooner than the latter ones, i.e, within the first 100 seconds. Second, the larger the ratio m/M is, the more efficient the ICG methods are compared to the other benchmarked methods.

6 Static ICG Iteration Complexities

This section establishes the iteration complexities for each of the static ICG methods in Section 3.

6.1 Static IA-ICG Iteration Complexity

This subsection establishes the key properties of the static IA-ICG method.

Lemma 8. *Let $\{(y_i, \hat{y}_i, \hat{v}_i)\}_{i=1}^k$ be the collection of iterates generated by the static IA-ICG method. For every $i \geq 1$, we have*

$$\frac{1}{4\lambda}\|y_{i-1} - y_i\|^2 \leq \phi(y_{i-1}) - \widetilde{\ell}_\phi(y_i; y_{i-1}) - \frac{M_1}{2}\|y_i - y_{i-1}\|^2 \leq \phi(y_{i-1}) - \phi(y_i), \tag{46}$$

where $\widetilde{\ell}_\phi$ is as in (29).

Proof. Let $i \geq 1$ be fixed and let $(y_i, v_i, \varepsilon_i)$ be the point output by the i^{th} successful call to the R-ACG algorithm. Moreover, let $\Delta_1(\cdot;\cdot,\cdot)$ be as in (10) with (ψ_s, ψ_n) given by (28). Using the definition of $\widetilde{\ell}_\phi$, step 2 of the method, and fact that $(y^a, v, \varepsilon) = (y_i, v_i, \varepsilon_i)$ solves Problem $\mathcal{B}$ in Section 2 with (μ, ψ_s, ψ_n) as in (28), we have that

$$\varepsilon_i \geq \Delta_1(y_{i-1}; y_i, v_i) = \lambda\widetilde{\ell}_\phi(y_i; y_{i-1}) - \lambda\phi(y_{i-1}) - \langle v_i, y_i - y_{i-1}\rangle + \|y_i - y_{i-1}\|^2.$$

Rearranging the above inequality and using assumption (A2), (27), and the fact that $\langle a, b\rangle \geq$

Parameters $(\theta, m, M)^T$	Time t	Minimum Subgradient Size ($\min_{i\le k}\|\hat{V}_i\|$)						
		ECG	AIPP	AG	UP	NCF	IA	DA
$\begin{bmatrix}1\\169\\201\end{bmatrix}$	100	392.4	500.3	501.2	506.0	482.5	**33.9**	75.5
	200	392.4	478.4	492.3	506.0	465.0	**33.9**	43.2
	400	392.4	182.2	455.9	57.1	407.0	**33.9**	43.2
	800	392.4	36.7	320.6	57.1	284.3	**33.9**	43.2
$\begin{bmatrix}0.1\\11443\\2001\end{bmatrix}$	100	489.1	505.9	505.7	-	505.8	**43.4**	416.0
	200	476.9	505.6	505.3	505.5	505.5	**43.4**	76.9
	400	449.5	503.4	503.1	505.5	505.0	**43.4**	53.8
	800	399.4	240.8	496.2	505.3	503.9	**43.4**	53.8
$\begin{bmatrix}0.01\\839400\\20001\end{bmatrix}$	100	505.6	505.9	505.8	505.9	505.9	**48.6**	137.5
	200	505.1	505.9	505.7	505.9	505.9	**48.6**	58.6
	400	504.1	498.1	504.9	505.9	505.9	**48.6**	58.6
	800	502.2	176.9	502.1	505.9	505.9	**48.6**	58.6

Table 4: Minimum subgradient sizes for the binomial dataset. Times are in seconds and "-" indicates a run that did not generate a subgradient within the given time limit.

$-\|a\|^2/2 - \|b\|^2/2$ for every $a, b \in \mathcal{Z}$ yields

$$\begin{aligned}
&\lambda\phi(y_{i-1}) - \lambda\widetilde{\ell}_\phi(y_i; y_{i-1}) \ge \langle v_i, y_{i-1} - y_i\rangle - \varepsilon_i + \|y_i - y_{i-1}\|^2 \\
&= \frac{1}{2}\|y_i - y_{i-1}\|^2 - \frac{1}{2}\left(\|v_i\|^2 + 2\varepsilon_i\right) \ge \left(\frac{1-\theta^2}{2}\right)\|y_i - y_{i-1}\|^2 \\
&= \frac{\lambda M_1}{2}\|y_i - y_{i-1}\|^2 + \left(\frac{1 - \lambda M_1 - \theta^2}{2}\right)\|y_i - y_{i-1}\|^2 \\
&= \frac{\lambda M_1}{2}\|y_i - y_{i-1}\|^2 + \frac{1}{4}\|y_i - y_{i-1}\|^2. \qquad (47)
\end{aligned}$$

Rearranging terms yields the first inequality of (46). The second inequality of (46) follows from the first inequality, the fact that $\widetilde{\ell}_\phi(y_i; y_{i-1}) + M_1\|y_i - y_{i-1}\|^2/2 \ge \phi(y_i)$ from assumption (A2), and the definition of $\widetilde{\ell}_\phi$. □

The next results establish the rate at which the residual $\|\hat{v}_i\|$ tends to 0.

Lemma 9. *Let $p > 1$ be given. Then, for every $a, b \in \mathbb{R}^k$, we have*

$$\min_{1\le i\le k}\{|a_i b_i|\} \le k^{-p}\,\|a\|_1\,\|b\|_{1/(p-1)}\,.$$

Proof. Let $p > 1$ and $a, b \in \mathbb{R}^k$ be fixed and let $q \ge 1$ be such that $p^{-1} + q^{-1} = 1$. Using the fact that $\langle x, y\rangle \le \|x\|_p\|y\|_q$ for every $x, y \in \mathbb{R}^k$, and denoting $\tilde{a}$ and $\tilde{b}$ to be vectors with entries $|a_i|^{1/p}$ and $|b_i|^{1/p}$, respectively, we have that

$$\begin{aligned}
&k\min_{1\le i\le k}\{|a_i b_i|\}^{1/p} \le \sum_{i=1}^k |a_i b_i|^{1/p} \\
&\le \|\tilde{a}\|_p\|\tilde{b}\|_q = \|a\|_1^{1/p}\left(\sum_{i=1}^k |b_i|^{q/p}\right)^{1/q} = \left(\|a\|_1\|b\|_{q/p}\right)^{1/p}.
\end{aligned}$$

Parameters $(\theta, m, M)^T$	Time t	Minimum Subgradient Size ($\min_{i\le k}\|\hat{V}_i\|$)						
		ECG	AIPP	AG	UP	NCF	IA	DA
$\begin{bmatrix}1\\169\\201\end{bmatrix}$	100	-	564.3	562.7	-	552.2	**39.1**	362.3
	200	433.5	551.8	554.1	566.6	536.2	**30.0**	80.3
	400	433.5	351.5	526.6	566.6	501.7	**30.0**	40.8
	800	433.5	35.6	433.7	55.8	435.7	**30.0**	40.8
$\begin{bmatrix}0.1\\11443\\2001\end{bmatrix}$	100	533.8	566.4	566.2	-	566.2	**41.0**	465.0
	200	507.4	566.1	565.7	566.0	566.0	**41.0**	81.4
	400	478.2	563.6	561.8	566.0	565.6	**41.0**	50.0
	800	417.6	159.0	549.9	565.8	564.4	**41.0**	50.0
$\begin{bmatrix}0.01\\839400\\20001\end{bmatrix}$	100	565.5	566.4	566.2	566.4	566.4	**45.8**	54.3
	200	564.6	563.9	565.5	566.4	566.4	**45.8**	54.3
	400	562.7	186.1	563.1	566.3	566.4	**45.8**	54.3
	800	559.1	143.6	555.6	566.3	566.3	**45.8**	54.3

Table 5: Minimum subgradient sizes for the truncated normal dataset. Times are in seconds and "-" indicates a run that did not generate a subgradient within the given time limit.

Dividing by k, taking the p^{th} power on both sides, and using the fact that $p/q = p-1$, yields

$$\min_{1\le i\le k}\{|a_i b_i|\} \le k^{-p}\|a\|_1\|b\|_{q/p} = k^{-p}\|a\|_1\|b\|_{1/(p-1)}.$$

□

Proposition 10. *Let $\{(y_i, \hat{y}_i, \hat{v}_i)\}_{i=1}^k$ be as in Lemma 8 and define the quantities*

$$\begin{aligned} L_{1,k}^{\text{avg}} &:= \frac{1}{k}\sum_{i=1}^k L_1(y_i, y_{i-1}), \quad C_{\lambda,k}^{\text{avg}} := \frac{1}{k}\sum_{i=1}^k C_\lambda(\hat{y}_i, y_i), \\ D_k^{\text{avg}} &:= L_{1,k}^{\text{avg}} + \frac{\theta}{\lambda}C_{\lambda,k}^{\text{avg}}, \quad \beta_1 := \left(\frac{1+\overline{C}_\lambda}{\lambda}\right) + \sqrt{2}\left(\frac{2+\lambda L_1 + \theta\overline{C}_\lambda}{\lambda}\right), \end{aligned} \tag{48}$$

where $C_\lambda(\cdot,\cdot)$ and $\overline{C}_\lambda$ are as in (26) and (29), respectively. Then, we have

$$\min_{i\le k}\|\hat{v}_i\| = \mathcal{O}_1\left(\left[\sqrt{\lambda}L_{1,k}^{\text{avg}} + \frac{1+\theta C_{\lambda,k}^{\text{avg}}}{\sqrt{\lambda}}\right]\left[\frac{\phi(z_0)-\phi_*}{k}\right]^{1/2}\right) + \frac{\hat{\rho}}{2}.$$

Proof. Using Lemma 3 with $(y, w) = (y_i, y_{i-1})$ and the fact that $C_\lambda(\cdot,\cdot) \le \overline{C}_\lambda$ and $L_1(\cdot,\cdot) \le L_1$, we have $\|\hat{v}_i\| \le \mathcal{E}_i\|y_i - y_{i-1}\|$, for every $i \le k$, where

$$\mathcal{E}_i := \frac{2+\lambda L_1(y_i, y_{i-1}) + \theta C_\lambda(\hat{y}_i, y_i)}{\lambda} \quad \forall i \ge 1.$$

As a consequence, using the sum of the second bound in Lemma 8 from $i=1$ to k, the definitions in (48), and Lemma 9 with $p = 3/2$, $a_i = \mathcal{E}_i$, and $b_i = \|y_i - y_{i-1}\|$ for $i = 1$ to k, yields

$$\begin{aligned} \min_{i\le k}\|\hat{v}_i\| &\le \min_{i\le k}\mathcal{E}_i\|y_i - y_{i-1}\| \le \frac{1}{k^{3/2}}\left(\sum_{i=1}^k \mathcal{E}_i\right)\left(\sum_{i=1}^k \|y_i - y_{i-1}\|^2\right)^{1/2} \\ &= \mathcal{O}_1\left(\left[\sqrt{\lambda}L_{1,k}^{\text{avg}} + \frac{1+\theta C_{\lambda,k}^{\text{avg}}}{\sqrt{\lambda}}\right]\left[\frac{\phi(z_0)-\phi_*}{k}\right]^{1/2}\right). \end{aligned} \tag{49}$$

□

We are now ready to give the proof of Theorem 4.

Proof of Theorem 4. (a) This follows from Proposition 10, the fact that $C_\lambda(\cdot,\cdot) \le \overline{C}_\lambda$ and $L_{f_1}(\cdot,\cdot) \le L_1$, and the stopping condition in step 3.

(b) The fact that $(\hat{y},\hat{v}) = (\hat{y}_k,\hat{v}_k)$ satisfies the inclusion of (22) follows from Lemma 3 with $(y,v,w) = (y_k,v_k,y_{k-1})$. The fact that $\|\hat{v}\| \le \hat{\rho}$ follows from the stopping condition in step 3.

(c) This follows from Proposition 2(c) and the fact that method stops in finite number of iterations from part (a). □

6.2 Static DA-ICG Iteration Complexity

This subsection establishes several key properties of static DA-ICG method.

To avoid repetition, we assume throughout this subsection that $k \ge 1$ denotes an arbitrary successful outer iteration of the DA-ICG method and let

$$\{(a_i, A_i, y_i, y_i^a, x_i, \tilde{x}_{i-1}, \hat{y}_i, \hat{v}_i, v_i, \varepsilon_i)\}_{i=1}^k$$

denote the sequence of all iterates generated by it up to and including the k^{th} iteration. Observe that this implies that the i^{th} DA-ICG outer iteration for any $1 \le i \le k$ is successful, i.e., the (only) R-ACG call in step 2 of the DA-ICG method does not stop with failure and $\Delta_1(y_{i-1}; y_i^a, v_i) \le \varepsilon_i$. Moreover, throughout this subsection we let

$$\widetilde{\gamma}_i(u) = \ell_{f_1}(u;\tilde{x}_{i-1}) + f_2(u) + h(u), \quad \gamma_i(u) = \widetilde{\gamma}_i(y_i^a) + \frac{1}{\lambda}\langle v_i + \tilde{x}_{i-1} - y_i^a, u - y_i^a\rangle. \tag{50}$$

The first set of results present some basic properties about the functions $\widetilde{\gamma}_i$ and γ_i as well as the iterates generated by the method.

Lemma 11. *Let $\Delta_1(\cdot;\cdot,\cdot)$ be as in (10) with (ψ_s,ψ_n) given by (28). Then, the following statements hold for any $s \in \operatorname{dom} h$ and $1 \le i \le k$:*

(a) $\gamma_i(y_i^a) = \widetilde{\gamma}_i(y_i^a)$;

(b) $x_i = \operatorname{argmin}_{u\in\Omega}\{\lambda a_{i-1}\gamma_i(u) + \|u - x_{i-1}\|^2/2\}$;

(c) $y_i^a - v_i = \operatorname{argmin}_{u\in\mathcal{Z}}\{\lambda\gamma_i(u) + \|u - \tilde{x}_{i-1}\|^2/2\}$;

(d) $-M_1\|u-\tilde{x}_{i-1}\|^2/2 \le \widetilde{\gamma}_i(u) - \phi(u) \le m_1\|u-\tilde{x}_{i-1}\|^2/2$;

(e) $\phi(y_{i-1}) \ge \phi(y_i)$ *and* $\phi(y_i^a) \ge \phi(y_i)$.

Proof. To keep the notation simple, denote

$$\begin{aligned}(y_+^a, y_+, y, \tilde{x}) &= (y_i^a, y_i, y_{i-1}, \tilde{x}_{i-1}), \quad (x_+, x) = (x_i, x_{i-1}),\\ (A_+, A, a) &= (A_i, A_{i-1}, a_{i-1}), \quad (v,\varepsilon) = (v_i,\varepsilon_i).\end{aligned} \tag{51}$$

(a) This is immediate from the definitions of γ and $\widetilde{\gamma}$ in (50).

(b) Define $\widehat{x}_i := x_{k-1} - a_{k-1}(v_k + \tilde{x}_{k-1} - y_k^a)$. Using the definition of γ in (50), we have that

$$\begin{aligned}&\operatorname*{argmin}_{u\in\Omega}\left\{\lambda a\gamma(u) + \frac{1}{2}\|u-x\|^2\right\} = \operatorname*{argmin}_{u\in\Omega}\left\{a\langle v + \tilde{x} - y_+^a, u - x\rangle + \frac{1}{2}\|u-x\|^2\right\}\\ &= \operatorname*{argmin}_{u\in\Omega}\frac{1}{2}\|u - (x - a[v + \tilde{x} - y_+^a])\|^2 = \operatorname*{argmin}_{u\in\Omega}\frac{1}{2}\|u - \widehat{x}_+\|^2 = x_+.\end{aligned}$$

(c) Using the definition of γ in (50), we have that

$$\lambda \nabla \gamma \left(y_+^a - v\right) + \left(y_+^a - v\right) - \tilde{x} = \left(v + \tilde{x} - y_+^a\right) + \left(y_+^a - v\right) - \tilde{x} = 0,$$

and hence, the point $y_+^a - v$ is the global minimum of $\lambda\gamma + \|\cdot - \tilde{x}\|^2/2$.

(d) This follows from inequality (21) with $i = 1$ and the definition of $\widetilde{\gamma}$ in (50).

(e) This follows immediately from the update rule of y_i in (36). □

Lemma 12. *Let $w = \tilde{x}_{i-1}$, the pair (ψ_n, ψ_s) be as in (28), and $\Delta_1(\cdot;\cdot,\cdot)$ be as in (10) with (ψ_s, ψ_n) given by (28). Then, following statements hold:*

(a) the triple $(y_i^a, v_i, \varepsilon_i)$ solves Problem $\mathcal{B}$ and satisfies $\Delta_1(y_{i-1}; y_i^a, v_i) \leq \varepsilon$, and hence

$$\|v_i\| + 2\varepsilon_i \leq \theta^2 \|y_i^a - \tilde{x}_{i-1}\|^2, \quad \Delta_1(u; y_i^a, v_i) \leq \varepsilon_i \quad \forall u \in \{\hat{y}_i, y_{i-1}\}, \tag{52}$$

(b) if f_2 is convex, then $(y_i^a, v_i, \varepsilon_i)$ solves Problem $\mathcal{A}$;

(c) $\Delta_1(s; y_i^a, v_i) = \lambda[\gamma_i(s) - \widetilde{\gamma}_i(s)]$;

(d) $\Delta_1(y_i; y_i^a, v_i) \leq \varepsilon$.

Proof. (a) This follows from step 2 of the DA-ICG method and Proposition 2(b).

(b) This follows from steps 2 and 3 of the DA-ICG method, the fact that h is convex, and Proposition 2(c) with $\psi_s = \widetilde{\gamma}_i + \|\cdot - \tilde{x}_{i-1}\|^2/2$.

(c) Using the definitions of (ψ_s, ψ_n) and $(\gamma, \widetilde{\gamma})$ in (28) and (50), respectively, we have that

$$\begin{aligned}
\Delta_1(s; y_+^a, v) &= (\psi_s + \psi_n)(y_+^a) - (\psi_s + \psi_n)(s) - \langle v, y_+^a - s\rangle + \frac{1}{2}\|s - y_+^a\|^2 \\
&= \left[\lambda\widetilde{\gamma}(y_+^a) + \frac{1}{2}\|y_+^a - \tilde{x}\|^2\right] - \left[\lambda\widetilde{\gamma}(s) + \frac{1}{2}\|s - \tilde{x}\|^2\right] - \langle v, y_+^a - s\rangle + \frac{1}{2}\|s - y_+^a\|^2 \\
&= \left[\lambda\gamma(s) + \frac{1}{2}\|s - \tilde{x}\|^2\right] - \left[\lambda\widetilde{\gamma}(s) + \frac{1}{2}\|s - \tilde{x}\|^2\right] = \lambda\gamma(s) - \lambda\widetilde{\gamma}(s).
\end{aligned}$$

(d) If $y_i = y_{i-1}$, then this follows from step 3 of the method. On the other hand, if $y_i = y_i^a$, then this follows from part (c). □

We now state (without proof) some well-known properties of A_i and a_{i-1}.

Lemma 13. *For every $1 \leq i \leq k$, we have that:*

(a) $a_{i-1}^2 = A_i$;

(b) $i^2/4 \leq A_i \leq i^2$.

The next two lemmas are technical results that are needed to establish the key inequality in Proposition 16.

Lemma 14. *For every $u \in \operatorname{dom} h$ and $1 \leq i \leq k$, we have that*

$$\frac{1}{2}\left(A_{i-1}\|y_{i-1} - \tilde{x}_{i-1}\|^2 + a_{i-1}\|u - \tilde{x}_{i-1}\|^2\right) \leq 2D_\Omega^2 + a_{i-1}D_h^2.$$

Proof. Throughout the proof, we use the notation in (51). Using the relation $(p+q)^2 \le 2p^2+2q^2$ for every $p,q\in\mathbb{R}$, Lemma 13(a), the fact that $A\le A^+$, $x\in\Omega$, and $y\in\operatorname{dom} h$, and the definitions of $\tilde{x}$ in (35) and of D_Ω and D_h in (37), we conclude that

$$\begin{aligned}
&A\|y-\tilde{x}\|^2+a\|u-\tilde{x}\|^2 = A\left\|\frac{a}{A_+}(y-x)\right\|^2+a\left\|\frac{A}{A_+}(u-y)+\frac{a}{A_+}(u-x)\right\|^2\\
&\le \frac{A}{A_+}\left(\|(y-u)+(u-x)\|^2+2a\left[\frac{A^2}{A_+^2}\|u-y\|^2+\frac{a^2}{A_+^2}\|u-x\|^2\right]\right)\\
&\le \frac{2A}{A^+}\left(\|u-y\|^2+\|u-x\|^2\right)+2a\|u-y\|^2+\frac{2a}{A_+}\|u-x\|^2\\
&\le 2\left[\|u-x\|^2+(1+a)\|u-y\|^2\right]\le 2[D_\Omega^2+(1+a)D_h^2].
\end{aligned}$$

The conclusion now follows from dividing both sides of the above inequalities by 2 and using the fact that $D_h\le D_\Omega$. □

Lemma 15. *For every $u\in\operatorname{dom} h$ and $1\le i\le k$, we have that*

$$\begin{aligned}
&A_i\left[\phi(y_i)+\left(\frac{1-\lambda M_1}{2\lambda}\right)\|y_i^a-\tilde{x}_{i-1}\|^2-\frac{\|v_i\|^2}{2\lambda}\right]+\frac{1}{2\lambda}\|u-x_i\|^2\\
&\le A_{i-1}\gamma_i(y_{i-1})+a_{i-1}\gamma_i(u)+\frac{1}{2\lambda}\|u-x_{i-1}\|^2.
\end{aligned}\tag{53}$$

Proof. Throughout the proof, we use the notation in (51). We first present two key expressions. First, using the definition of γ in (50) and Lemma 11(c), it follows that

$$\begin{aligned}
&\min_{u\in\mathcal{Z}}\left\{\lambda\gamma(u)+\frac{1}{2}\|u-\tilde{x}\|^2\right\}=\lambda\widetilde{\gamma}(y_+^a)-\langle v+\tilde{x}-y_+^a,v\rangle+\frac{1}{2}\|v+\tilde{x}-y_+^a\|^2\\
&=\lambda\widetilde{\gamma}(y_+^a)-\|v\|^2-\langle v,\tilde{x}-y_+^a\rangle+\frac{1}{2}\|v+\tilde{x}-y_+^a\|^2\\
&=\lambda\widetilde{\gamma}(y_+^a)-\frac{1}{2}\|v\|^2+\frac{1}{2}\|\tilde{x}-y_+^a\|^2.
\end{aligned}\tag{54}$$

Second, Lemma 11(b) and the fact that the function $a\gamma+\|\cdot-x\|^2/(2\lambda)$ is $(1/\lambda)$–strongly convex imply that

$$a\gamma(x_+)+\frac{1}{2\lambda}\|x_+-x\|^2\le a\gamma(u)+\frac{1}{2\lambda}\|u-x\|^2-\frac{1}{2\lambda}\|u-x_+\|^2.\tag{55}$$

Using (54), Lemma 11(d)–(e), Lemma 13(a), and the fact that γ is affine, we have that

$$\begin{aligned}
&A_+\left[\phi(y_+)+\left(\frac{1-\lambda M_1}{2\lambda}\right)\|y_+^a-\tilde{x}\|^2\right]\le A_+\left[\widetilde{\gamma}(y_+^a)+\frac{1}{2\lambda}\|y_+^a-\tilde{x}\|^2\right]\\
&=A_+\left[\min_{u\in\mathcal{Z}}\left\{\gamma(u)+\frac{1}{2\lambda}\|u-\tilde{x}\|^2\right\}+\frac{\|v\|^2}{2\lambda}\right]\\
&\le A_+\left[\gamma\left(\frac{Ay+ax_+}{A_+}\right)+\frac{1}{2\lambda}\left\|\frac{Ay+ax_+}{A_+}-\frac{Ay+ax}{A_+}\right\|^2+\frac{\|v\|^2}{2\lambda}\right]\\
&=A\gamma(y)+a\gamma(x_+)+\frac{a^2}{2\lambda A_+}\|x-x_+\|^2+\frac{A_+}{2\lambda}\|v\|^2\\
&=A\gamma(y)+a\gamma(x_+)+\frac{1}{2\lambda}\|x-x_+\|^2+\frac{A_+}{2\lambda}\|v\|^2
\end{aligned}\tag{56}$$

The conclusion now follows from combining (55) with (56). □

We now present an inequality that plays an important role in the analysis of the DA-ICG method.

Proposition 16. *Let $\Delta_1(\cdot;\cdot,\cdot)$ be as in (10) with (ψ_s, ψ_n) as in (28), and define*

$$\theta_i(u) := A_i\left[\phi(y_i) - \phi(u)\right] + \frac{1}{2\lambda}\|u - x_i\|^2 \quad \forall i \geq 0. \tag{57}$$

For every $u \in \operatorname{dom} h$ satisfying $\Delta_1(u; y_i^a, v_i) \leq \varepsilon$ and $1 \leq i \leq k$, we have that

$$\frac{A_i}{4\lambda}\|y_i^a - \tilde{x}_{i-1}\|^2 \leq m_1^+\left(a_{i-1}D_h^2 + 2D_\Omega^2\right) + \theta_{i-1}(u) - \theta_i(u). \tag{58}$$

Proof. Throughout the proof, we use the notation in (51) together with the notation $\pi = \pi_{i-1}$ and $\pi_+ = \pi_i$. Let $u \in \operatorname{dom} h$ be such that $\Delta_1(u; y_+^a, v) \leq \varepsilon$. Subtracting $A\phi(u)$ from both sides of the inequality in (53) and using the definition of π_+ we have

$$\begin{aligned}
&\frac{A_+}{2\lambda}\left[(1-\lambda M_1)\|y_+^a - \tilde{x}\|^2 - \|v\|^2\right] + \pi_+(u)\\
&= \frac{A_+}{2\lambda}\left[(1-\lambda M_1)\|y_+^a - \tilde{x}\|^2 - \|v\|^2\right] + A_+\left[\phi(y_+) - \phi(u)\right] + \frac{1}{2\lambda}\|u - y_+^a\|^2\\
&\leq A\gamma(y) + a\gamma(u) - A\phi(u) + \frac{1}{2\lambda}\|u - x\|^2\\
&= a\left[\gamma(u) - \phi(u)\right] + A\left[\gamma(y) - \phi(y)\right] + \pi(u).
\end{aligned} \tag{59}$$

Moreover, using Lemma 12(a) and (c), and with our assumption that $\Delta_1(u; y_+^a, v) \leq \varepsilon$, we have that

$$\gamma(s) - \phi(s) = \tilde{\gamma}(s) - \phi(s) + \frac{\Delta_1(s; y_+^a, v)}{\lambda} \leq \frac{m_1^+}{2}\|s - \tilde{x}\|^2 + \frac{\varepsilon}{\lambda} \quad \forall s \in \{u, y\}. \tag{60}$$

Combining (59), (60), and Lemma 14 then yields

$$\begin{aligned}
&\frac{A_+}{2\lambda}\left[(1-\lambda M_1)\|y_+^a - \tilde{x}\|^2 - \|v\|^2\right] + \pi_+(u)\\
&\leq \frac{m_1^+}{2}\left[a\|u - \tilde{x}\|^2 + A\|y - \tilde{x}\|^2\right] + \frac{\varepsilon A_+}{\lambda} + \pi(u) \leq m_1^+\left(aD_h^2 + 2D_\Omega^2\right) + \frac{\varepsilon A_+}{\lambda} + \pi(u).
\end{aligned}$$

Re-arranging the above terms and using (34) together with the first inequality in (52), we conclude that

$$\begin{aligned}
&m_1^+\left(aD_h^2 + 2D_\Omega^2\right) + \pi(u) - \pi_+(u) \geq \frac{A_+}{2\lambda}\left[(1-\lambda M_1)\|y_+^a - \tilde{x}\|^2 - \|v\|^2 - 2\varepsilon\right]\\
&\geq \frac{A_+(1 - \lambda M_1 - \theta^2)}{2\lambda}\|y_+^a - \tilde{x}\|^2 \geq \frac{A_+}{4\lambda}\|y_+^a - \tilde{x}\|^2.
\end{aligned}$$

□

The following result describes some important technical bounds obtained by summing (58) for two different choices of u (possibly changing with i) from $i = 1$ to k.

Proposition 17. *Let Δ_ϕ^0 and d_0 be as in (37) and define*

$$S_k := \frac{1}{4\lambda}\sum_{i=1}^{k} A_i\|y_i^a - \tilde{x}_{i-1}\|^2. \tag{61}$$

Then, the following statements hold:

(a) $S_k = \mathcal{O}_1(k^2[m_1^+ D_h^2 + \Delta_\phi^0] + k[m_1^+ + 1/\lambda]D_\Omega^2)$;

(b) if f_2 *is convex, then* $S_k = \mathcal{O}_1(k^2 m_1^+ D_h^2 + k m_1^+ D_\Omega^2 + d_0^2/\lambda)$.

Proof. (a) Let $\Delta_1(\cdot;\cdot,\cdot)$ be defined as in (10) with (ψ_s, ψ_n) given by (28). Using (57), the fact that $x_i, y_i^a \in \Omega$, the fact that A_i is nonnegative and increasing, and the definitions of θ_i and D_Ω in (57) and (37), respectively, we have that

$$\begin{aligned}\sum_{i=1}^{k} [\theta_{i-1}(y_i) - \theta_i(y_i)] &\leq \sum_{i=1}^{k} A_{i-1} [\phi(y_{i-1}) - \phi(y_i)] + \frac{1}{2\lambda} \sum_{i=1}^{k} \|y_i - x_{i-1}\|^2 \\ &\leq A_k \sum_{i=1}^{k} [\phi(y_{i-1}) - \phi(y_i)] + \frac{k}{2\lambda} D_\Omega^2 \leq A_k [\phi(y_0) - \phi_*] + \frac{k}{2\lambda} D_\Omega^2. \end{aligned} \tag{62}$$

Moreover, noting Lemma 12(d) and using Proposition 16 with $u = y_i$, we conclude that (58) holds with $u = y_i$ for every $1 \leq i \leq k$. Summing these k inequalities and using (62), the definition of S_k in (61), and Lemma 13(b) yields the desired conclusion.

(b) Assume now that f_2 is convex and let y_* be a point such that $\phi(y_*) = \phi_*$ and $\|y_0 - y_*\| = d_0$. It then follows from Lemma 12(b) and Proposition 1(d) with $(y, v) = (y_i^a, v_i)$ that $\Delta_1(y_*; y_i^a, v_i) \leq \varepsilon$ for every $1 \leq i \leq k$. The conclusion now follows by using an argument similar to the one in (a) but which instead sums (58) with $u = y_*$ from $i = 1$ to k, and uses the fact that

$$\sum_{i=1}^{k} [\theta_{i-1}(y_*) - \theta_i(y_*)] = \theta_0(y_*) - \theta_k(y_*) \leq \frac{1}{2\lambda}\|y_0 - y_*\|^2 = \frac{d_0}{2\lambda},$$

where the inequality is due to the fact that $\theta_k(y_*) \geq 0$ (see (57)) and $A_0 = 0$. □

We now establish the rate at which the residual $\|\hat{v}_i\|$ tends to 0.

Proposition 18. *Let* S_k *be as in* (61). *Moreover, define the quantities*

$$\begin{aligned} L_{1,k}^{\text{avg}} &:= \frac{1}{k}\sum_{i=1}^{k} L_1(y_i^a, \tilde{x}_{i-1}), \quad C_{\lambda,k}^{\text{avg}} := \frac{1}{k}\sum_{i=1}^{k} C_\lambda(\hat{y}_i, y_i^a), \\ D_k^{\text{erg}} &:= L_{1,k}^{\text{erg}} + \frac{\theta}{\lambda} C_{\lambda,k}^{\text{erg}}, \quad 8\sqrt{2}\left(\frac{2 + \lambda L_1 + \theta \overline{C}_\lambda}{\lambda}\right), \end{aligned} \tag{63}$$

where $C_\lambda(\cdot,\cdot)$ *and* $\overline{C}_\lambda$ *are as in* (26) *and* (29), *respectively. Then, we have*

$$\min_{i \leq k} \|\hat{v}_i\| = \mathcal{O}_1\left(\left[\sqrt{\lambda} L_{1,k}^{\text{avg}} + \frac{1 + \theta C_{\lambda,k}^{\text{avg}}}{\sqrt{\lambda}}\right]\left[\frac{S_k}{k^3}\right]^{1/2}\right) + \frac{\hat{\rho}}{2}.$$

Proof. Let $\ell = \lceil k/2 \rceil$. Using Lemma 3 with $(z, w) = (y_i^a, \tilde{x}_{i-1})$ and the bounds $C_\lambda(\cdot,\cdot) \leq \overline{C}_\lambda$ and $L_1(\cdot,\cdot) \leq L_1$ we have that $\|\hat{v}_i\| \leq \mathcal{E}_i \|y_i^a - \tilde{x}_{i-1}\|$, for every $\ell \leq i \leq k$, where

$$\mathcal{E}_i = \frac{2 + \lambda L_1(y_i^a, \tilde{x}_{i-1}) + \theta C_\lambda(\hat{y}_i, y_i^a)}{\lambda} \quad \forall i \geq 1.$$

As a consequence, using the definition of S_k in (61), the definitions in (63), Lemma 9 with $p = 3/2$, $a_i = \mathcal{E}_i/\sqrt{A_i}$, and $b_i = \sqrt{A_i}\|y_i^a - \tilde{x}_{i-1}\|$ for $i \in \{\ell, ..., k\}$, Lemma 13(b), and the fact that

$(k-\ell+1) \geq k/2$, yields

$$\begin{aligned}
&\min_{\ell \leq i \leq k} \|\hat{v}_i\| \leq \min_{\ell \leq i \leq k} \mathcal{E}_i \|y_i^a - \tilde{x}_{i-1}\| \\
&\leq \frac{1}{(k-\ell+1)^{3/2}} \left(\sum_{i=\ell}^{k} \frac{\mathcal{E}_i}{\sqrt{A_i}} \right) \left(\sum_{i=\ell}^{k} A_i \|y_i^a - \tilde{x}_{i-1}\|^2 \right)^{1/2} \\
&\leq \frac{2^{3/2}}{k^{3/2}} \left(\frac{2}{k} \sum_{i=1}^{k} \mathcal{E}_i \right) (4\lambda S_k)^{1/2} = \mathcal{O}_1 \left(\left[\sqrt{\lambda} L_{1,k}^{\text{avg}} + \frac{1+\theta C_{\lambda,k}^{\text{avg}}}{\sqrt{\lambda}} \right] \left[\frac{S_k}{k^3} \right]^{1/2} \right).
\end{aligned}$$

□

We are now ready to prove Theorem 5.

Proof of Theorem 5. (a) This follows from Proposition 18, Proposition 17(a), the fact that $C_\lambda(\cdot,\cdot) \leq \overline{C}_\lambda$ and $L_{f_1}(\cdot,\cdot) \leq L_1$, and the termination condition in step 4.
(b) The fact that $(\hat{y}, \hat{v}) = (\hat{y}_k, \hat{v}_k)$ satisfies the inclusion of (22) follows from Lemma 3 with $(y, v, z_0) = (y_k^a, v_k, \tilde{x}_{k-1})$. The fact that $\|\hat{v}\| \leq \hat{\rho}$ follows from the stopping condition in step 4.

(c) The fact that the method does not fail follows from Proposition 2(c). The bound in (39) follows from a similar argument as in part (a) except that Proposition 17(a) is replaced with Proposition 17(b). □

A Technical Bounds

The result below presents a basic property of the composite gradient step.

Proposition 19. *Let $h \in \overline{\text{Conv}}\,(\mathcal{Z})$, $z \in \operatorname{dom} h$, and g be a differentiable function on $\operatorname{dom} h$ which satisfies $g(u) - \ell_g(u;z) \leq L\|u-z\|^2/2$ for some $L \geq 0$ and every $u \in \operatorname{dom} g$. Moreover, define*

$$\hat{z} := \operatorname*{argmin}_u \left\{ \ell_g(u;z) + h(u) + \frac{L}{2}\|u-z\|^2 \right\}.$$

Then, it holds that

$$\frac{L}{2}\|z-\hat{z}\|^2 \leq (g+h)(z) - (g+h)(\hat{z}).$$

Proof. Using the definition of $\hat{z}$, the fact that $\ell_g(\cdot;z) + h(\cdot) + L\|\cdot - z\|^2/2$ is L-strongly convex, and the assumed bound $g(u) - \ell_g(u;z) \leq L\|u-z\|^2/2$ at $u = \hat{z}$, we have

$$(g+h)(z) = \ell_g(z;z) + h(z) \geq \ell_g(\hat{z};z) + h(\hat{z}) + L\|\hat{z}-z\|^2 \geq (g+h)(\hat{z}) + \frac{L}{2}\|\hat{z}-z\|^2.$$

□

B R-ACG Algorithm

This section presents technical results related to the R-ACG algorithm.

The first set of results describes some basic properties of the generated iterates.

Proposition 20. *If ψ_s is μ–strongly convex, then the following statements hold:*

(a) $z_j^c = \operatorname{argmin}_{u\in\mathcal{Z}} \{B_j\Gamma_j(u) + \|u - z_0^c\|^2/2\}$;

(b) $\Gamma_j \le \psi$ *and* $B_j\psi(z_j) \le \inf_{u\in\mathcal{Z}} \{B_j\Gamma_j(u) + \|u - z_0^c\|^2/2\}$;

(c) $\eta_j \ge 0$ *and* $r_j \in \partial_{\eta_j} \left(\psi - \mu\|\cdot - z_j\|^2/2\right)(z_j)$;

(d) it holds that

$$\left(\frac{1}{1+\mu B_j}\right)\|B_j r_j + z_j - z_0\|^2 + 2B_j\eta_j \le \|z_j - z_0\|^2$$

Proof. (a) See [15, Proposition 1].

(b) See [15, Proposition 1(b)].

(c) The optimality of z_j^c in part (a), the μ-strong convexity of Γ_j, and the definition of r_j imply that

$$\begin{aligned} r_j &= \frac{z_0^c - z_j^c}{B_j} + \mu(z_j - z_j^c) \in \partial\left(\Gamma_j - \frac{\mu}{2}\|\cdot - z_j^c\|^2 + \mu\left\langle \cdot, z_j^c - z_j\right\rangle\right)(z_j^c) \\ &= \partial\left(\Gamma_j - \frac{\mu}{2}\|\cdot - z_j\|^2\right)(z_j^c). \end{aligned}$$

Using the above inclusion, the definition of η_j, the fact that $\Gamma_j - \mu\|\cdot\|^2/2$ is affine, and part (b), we now conclude that

$$\begin{aligned} \psi(z) - \frac{\mu}{2}\|z - z_j\|^2 \ge \Gamma_j(z) - \frac{\mu}{2}\|z - z_j\|^2 &= \Gamma_j(z_j^c) - \frac{\mu}{2}\|z_j^c - z_j\|^2 + \left\langle r_j, z - z_j^c\right\rangle \\ &= \psi(z_j) + \langle r_j, z - z_j\rangle - \eta_j, \end{aligned}$$

for every $z \in \operatorname{dom}\psi_n$, which is exactly the desired inclusion. The fact that $\eta_j \ge 0$ follows from the above inequality with $z = z_j$.

(d) It follows from parts (a)–(b) and the definition of η_j that

$$\begin{aligned} \eta_j &\le \Gamma_j(u) + \frac{1}{2B_j}\|u - z_0\|^2 - \psi(z_j) \\ &= \frac{\mu}{2}\|z_j - z_j^c\|^2 - \frac{1}{B_j}\left\langle z_0 - z_j^c, z_j - z_j^c\right\rangle + \frac{1}{2B_j}\|z_j^c - z_0\|^2 \\ &= \frac{1}{2B_j}\|z_j - z_0\|^2 - \frac{1}{2B_j}(1+\mu B_j)\|z_j - z_j^c\|^2 \\ &= \frac{1}{2B_j}\|z_j - z_0\|^2 - \frac{1}{2B_j(1+\mu B_j)}\|B_j r_j + z_j - z_0\|^2. \end{aligned}$$

Multiplying both sides of the above inequality by $2B_j$ yields the desired conclusion. □

The next result presents the general iteration complexity of the algorithm, i.e. Proposition 2(a).

Proof of Proposition 2(a). Let ℓ be the first iteration where

$$\min\left\{\frac{B_\ell^2}{4(1+\mu B_\ell)}, \frac{B_\ell}{2}\right\} \ge K_\theta^2 \tag{64}$$

and suppose that the R-ACG has not stopped with failure before iteration ℓ. We show that it must stop with success at the end of the ℓ^{th} iteration. Combining the triangle inequality, the successful

check in step 3 of the method, (64), and the relation $(a+b)^2 \leq 2a^2+2b^2$ for all $a, b \in \mathbb{R}$, we first have that

$$\begin{aligned}
&\|r_\ell\|^2 + 2\eta_\ell \\
&\leq \max\left\{\frac{1+\mu B_\ell}{A_\ell^2}, \frac{1}{2B_\ell}\right\}\left(\frac{1}{1+\mu B_\ell}\|B_\ell r_\ell\|^2 + 4B_\ell\eta_\ell\right) \\
&\leq \max\left\{\frac{1+\mu B_\ell}{B_\ell^2}, \frac{1}{2B_\ell}\right\}\left(\frac{2}{1+\mu B_\ell}\|B_\ell r_\ell + z_\ell - z_0\|^2 + 2\|z_\ell - z_0\|^2 + 4B_\ell\eta_\ell\right) \\
&\leq \max\left\{\frac{4(1+\mu B_\ell)}{B_\ell^2}, \frac{2}{B_\ell}\right\}\|z_\ell - z_0\|^2 \leq \frac{1}{K_\theta^2}\|z_\ell - z_0\|^2 \leq \theta^2\|z_\ell - z_0\|^2,
\end{aligned}$$

and hence the method must terminate at the ℓ^{th} iteration. We now bound ℓ based on the requirement in (64). Solving for the quadratic in B_ℓ in the first bound of (64), it is easy to see that $B_\ell \geq 4\mu K_\theta^2 + 2K_\theta$ implies (64). On the other hand, for the second condition in (64), it is immediate that $B_\ell \geq 2K_\theta^2$ implies (64). In view of (18) and the previous two bounds, it follows that

$$B_\ell \geq \frac{1}{L}\left(1+\sqrt{\frac{\mu}{4L}}\right)^{2(\ell-1)} \geq 2K_\theta(1+2\mu K_\theta^2)$$

implies (64). Using the bound $\log(1+t) \geq t/(1+t)$ for $t \geq 0$ and the above bound on ℓ, it is straightforward to see that ℓ is on the same order of magnitude as in (19). □

C Refined ICG Points

This appendix presents technical results related to the refined points of the ICG methods.

The result below proves Lemma 3 from the main body of the paper.

Proof of Lemma 3. (a) Using Proposition 1(a), the definition of $\hat{v}$, and the definitions of ψ_s and ψ_n in (28), we have that

$$\begin{aligned}
\hat{v} &\in \frac{1}{\lambda}\left[\nabla\psi_s(\hat{y}) + \partial\psi_n(\hat{y}) + w - y\right] + \nabla f_1(\hat{y}) - \nabla f_1(w) \\
&= \frac{1}{\lambda}\left[\lambda\nabla f_1(w) + \lambda f_2(\hat{y}) + (w-y) + \lambda\partial h(y)\right] + \nabla f_1(\hat{y}) - \nabla f_1(w) \\
&= \nabla f_1(\hat{y}) + \nabla f_2(\hat{y}) + \partial h(\hat{y}),
\end{aligned}$$

(b) Using assumption (A3), Proposition 1(b), the choice of M in (28), and the fact that $\Delta_\mu(y_r; y, v) \leq \varepsilon$, we first observe that

$$\begin{aligned}
&\|\nabla f_1(\hat{y}) - \nabla f_1(z_0)\| - L_1(y, z_0)\|y - z_0\| \leq L_1(y,\hat{y})\|\hat{y} - y\| \\
&\leq \frac{L_1(y,\hat{y})\sqrt{2\Delta_\mu(y_r; y, v)}}{\sqrt{\lambda M_2^+ + 1}} \leq \frac{\theta L_1(y,\hat{y})}{\sqrt{\lambda M_2^+ + 1}}\|y - z_0\|. \qquad (65)
\end{aligned}$$

Using now (65), the choice of M in (28), Proposition 1(c) with $L(\cdot,\cdot) = \lambda L_2(\cdot,\cdot)$, the fact that

$\sigma \leq 1$, and the definition of $C_\lambda(\cdot,\cdot)$, we conclude that

$$\begin{aligned}\|\hat{v}\| &\leq \frac{1}{\lambda}\|v_r\| + \frac{1}{\lambda}\|y - z_0\| + \|\nabla f_1(\hat{y}) - \nabla f_1(z_0)\| \\ &\leq \left[L_1(y,z_0) + \frac{1+\theta}{\lambda} + \frac{\theta\left[\lambda M_2^+ + 1 + \lambda L_1(y,\hat{y}) + \lambda L_2(y,\hat{y})\right]}{\lambda\sqrt{\lambda M_2^+ + 1}}\right]\|y - z_0\| \\ &\leq \left[L_1(y,z_0) + \frac{2 + \theta C_\lambda(y,\hat{y})}{\lambda}\right]\|y - z_0\|.\end{aligned}$$

□

D Spectral Functions

This section presents some results about spectral functions as well as the proof of Propositions 6. It is assumed that the reader is familiar with the key quantities given in Subsection 4.1 (e.g., see (40) and (41)).

We first state two well-known results [2, 11] about spectral functions.

Lemma 21. *Let $\Psi = \Psi^{\mathcal{V}} \circ \sigma$ for some absolutely symmetric function $\Psi^{\mathcal{V}} : \mathbb{R}^r \mapsto \mathbb{R}$. Then, the following properties hold:*

(a) $\Psi^* = (\Psi^{\mathcal{V}} \circ \sigma)^* = (\Psi^{\mathcal{V}})^* \circ \sigma$;

(b) $\nabla\Psi = (\nabla\Psi^{\mathcal{V}}) \circ \sigma$;

Lemma 22. *Let $(\Psi, \Psi^{\mathcal{V}})$ be as in Lemma 21, the pair $(S,Z) \in \mathcal{Z} \times \operatorname{dom}\Psi$ be fixed, and the decomposition $S = P[\operatorname{dg}\sigma(S)]Q^*$ be an SVD of S, for some $(P,Q) \in \mathcal{U}^m \times \mathcal{U}^n$. If $\Psi \in \overline{\operatorname{Conv}}\,\mathbb{R}^{m\times n}$ and $\Psi^{\mathcal{V}} \in \overline{\operatorname{Conv}}\,\mathbb{R}^r$, then for every $M > 0$, we have*

$$S \in \partial\left(\Psi + \frac{M}{2}\|\cdot\|_F^2\right)(Z) \iff \begin{cases}\sigma(S) \in \partial\left(\Psi^{\mathcal{V}} + \frac{M}{2}\|\cdot\|^2\right)(\sigma(Z)), \\ Z = P[\operatorname{dg}\sigma(Z)]Q^*.\end{cases}$$

We now present a new result about spectral functions.

Theorem 23. *Let $(\Psi, \Psi^{\mathcal{V}})$ be as in Lemma 21 and the point $Z \in \mathbb{R}^{m\times n}$ be such that $\sigma(Z) \in \operatorname{dom}\Psi^{\mathcal{V}}$. Then for every $\varepsilon \geq 0$, we have $S \in \partial_\varepsilon\Psi(Z)$ if and only if $\sigma(S) \in \partial_{\varepsilon(S)}\Psi^{\mathcal{V}}(\sigma(Z))$, where*

$$\varepsilon(S) := \varepsilon - [\langle\sigma(Z),\sigma(S)\rangle - \langle Z,S\rangle] \geq 0. \tag{66}$$

Moreover, if S and Z have a simultaneous SVD, then $\varepsilon(S) = \varepsilon$.

Proof. Using Lemma 21(a), (66), and the well-known fact that $S \in \partial_\varepsilon\Psi(Z)$ if and only if $\varepsilon \geq \Psi(Z) + \Psi^*(S) - \langle Z,S\rangle$, we have that $S \in \partial_\varepsilon\Psi(Z)$ if and only if

$$\begin{aligned}\varepsilon(S) &= \varepsilon - [\langle\sigma(Z),\sigma(S)\rangle - \langle Z,S\rangle] \\ &\geq \Psi(Z) + \Psi^*(S) - \langle Z,S\rangle - [\langle\sigma(Z),\sigma(S)\rangle - \langle Z,S\rangle] \\ &= \Psi^{\mathcal{V}}(\sigma(Z)) + (\Psi^{\mathcal{V}})^*(\sigma(S)) - \langle\sigma(Z),\sigma(S)\rangle\,,\end{aligned}$$

or, equivalently, $\sigma(S) \in \partial_{\varepsilon(S)}\Psi^{\mathcal{V}}(\sigma(Z))$ and $\varepsilon(S) \geq 0$. To show that the existence of a simultaneous SVD of S and Z implies $\varepsilon(S) = \varepsilon$ it suffices to show that $\langle\sigma(S),\sigma(Z)\rangle = \langle S,Z\rangle$. Indeed, if $S = P[\operatorname{dg}\sigma(S)]Q^*$ and $Z = P[\operatorname{dg}\sigma(Z)]Q^*$, for some $(P,Q) \in \mathcal{U}^m \times \mathcal{U}^n$, then we have

$$\langle S,Z\rangle = \langle\operatorname{dg}\sigma(S), P^*P[\operatorname{dg}\sigma(Z)]Q^*Q\rangle = \langle\operatorname{dg}\sigma(S),\operatorname{dg}\sigma(Z)\rangle = \langle\sigma(S),\sigma(Z)\rangle.$$

□

Acknowledgments

The authors would like to thank the two anonymous referees and the associate editor for their insightful comments on earlier drafts of this paper.

References

[1] Miju Ahn, Jong-Shi Pang, and Jack Xin. Difference-of-convex learning: directional stationarity, optimality, and sparsity. *SIAM Journal on Optimization*, 27(3):1637–1665, 2017. 1

[2] Amir Beck. *First-order methods in optimization*, volume 25. SIAM, 2017. D

[3] Emmanuel J. Candes, Yonina C. Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015. 1

[4] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018. 1

[5] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1-2):503–558, 2019. 1

[6] Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, 156:59–99, 2016. 1, 3.3, 5.1

[7] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Generalized Uniformly Optimal Methods for Nonlinear Programming. *arXiv e-prints*, page arXiv:1508.07384, August 2015. 1, 5.1

[8] Yunlong He and Renato D. C. Monteiro. An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM J. Optim.*, 26(1):29–56, 2016. 2

[9] Weiwei Kong, Jefferson G. Melo, and Renato 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. 1

[10] Weiwei Kong, Jefferson G. Melo, and Renato D. C. Monteiro. An efficient adaptive accelerated inexact proximal point method for solving linearly constrained nonconvex composite problems. *Comp. Opt. and Appl.*, 76(2):305–346, 2020. 1, 3.2, 5.1

[11] Adrian S. Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2(1):173–183, 1995. D

[12] Jiaming Liang and Renato D. C. Monteiro. A Doubly Accelerated Inexact Proximal Point Method for Nonconvex Composite Optimization Problems. *arXiv e-prints*, page arXiv:1811.11378, November 2018. 1

[13] Jiaming Liang, Renato D. C. Monteiro, and Chee-Khian Sim. A FISTA-type accelerated gradient algorithm for solving smooth nonconvex composite optimization problems. *arXiv e-prints*, page arXiv:1905.07010, May 2019. 1, 5.1

[14] Renato D. C. Monteiro, Camilo Ortiz, and Benar F. Svaiter. Gradient methods for minimizing composite functions. *Math. Program.*, pages 1–37, 2012. 3.2

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

[16] Yurii Nesterov. Gradient methods for minimizing composite functions. *Math. Program.*, 2012. 5.1

[17] Courtney Paquette, Hongzhou Lin, Dmitriy Drusvyatskiy, Julien Mairal, and Zaid Harchaoui. Catalyst Acceleration for Gradient-Based Non-Convex Optimization. *arXiv e-prints*, page arXiv:1703.10993, March 2017. 1

[18] Tingni Sun and Cun-Hui Zhang. Calibrated elastic regularization in matrix completion. In *Advances in Neural Information Processing Systems*, pages 863–871, 2012. 5.2.1

[19] Bo Wen, Xiaojun Chen, and Ting Kei Pong. A proximal difference-of-convex algorithm with extrapolation. *Computational optimization and applications*, 69(2):297–324, 2018. 1

[20] Fei Wen, Rendong Ying, Peilin Liu, and Robert C. Qiu. Robust pca using generalized nonconvex regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 1

[21] Quanming Yao and James T. Kwok. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. *The Journal of Machine Learning Research*, 18(1):6574–6625, 2017. 1, 1, 5.2.1