Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Dosti, Endrit; Vorobyov, Sergiy A.; Charalambous, Themistoklis

A new class of composite objective multistep estimating sequence techniques

Published in:
Signal Processing

DOI:
10.1016/j.sigpro.2022.108889

Published: 01/05/2023

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
Dosti, E., Vorobyov, S. A., & Charalambous, T. (2023). A new class of composite objective multistep estimating sequence techniques. *Signal Processing*, *206*, Article 108889. https://doi.org/10.1016/j.sigpro.2022.108889

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Signal Processing 206 (2023) 108889

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A new class of composite objective multistep estimating sequence techniques

Endrit Dosti [a], Sergiy A. Vorobyov [a,*], Themistoklis Charalambous [a,b]

[a] School of Electrical Engineering, Aalto University, Espoo, Finland
[b] Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

ARTICLE INFO

Article history:
Received 22 April 2022
Revised 31 October 2022
Accepted 12 December 2022
Available online 16 December 2022

Keywords:
Accelerated first-order methods
Large-scale optimization
Composite objective
Estimating sequence
Gradient mapping
Line-search

ABSTRACT

A plethora of problems arising in signal processing, machine learning and statistics can be cast as large-scale optimization problems with a composite objective structure. Such problems are typically solved by utilizing iterative first-order algorithms. In this work, we devise a new accelerated gradient-based estimating sequence technique for solving large-scale optimization problems with composite objective structure. Specifically, we introduce a new class of estimating functions, which are obtained by utilizing both a tight lower bound on the objective function, as well as the gradient mapping technique. Then, using the proposed estimating functions, we construct a class of Composite Objective Multi-step Estimating-sequence Techniques (COMET), which are endowed with an efficient line-search procedure. We prove that our proposed COMET enjoys the accelerated convergence rate, and our newly established convergence results allow for step-size adaptation. Our theoretical findings are supported by extensive computational experiments on various problem types and real-world datasets. Moreover, our numerical results show evidence of the robustness of the proposed method to the imperfect knowledge of the smoothness and strong convexity parameters.

© 2022 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

In this work, we devise accelerated black-box methods for solving large-scale convex optimization problems with a composite objective structure by using only first-order information. The typical structure of such problems is

$$\underset{x\in\mathcal{R}^n}{\text{minimize}} \quad F(x) = f(x) + \tau g(x), \tau > 0, \tag{1}$$

where the function $f : \mathcal{R}^n \to \mathcal{R}$ is an L_f-smooth and μ_f-strongly convex function with $0 \le \mu_f \le L_f$. The regularizer $g : \mathcal{R}^n \to \mathcal{R}$ is a simple convex lower semi-continuous function with strong convexity parameter μ_g. Typically, in signal processing applications, the function $g(x)$ is "simple", meaning that a closed-form solution for minimizing the summation of g and some auxiliary functions can be easily found [1]. In more practical terms, the assumption on the simplicity of g implies that its proximal map, defined as

$$\text{prox}_{\tau g} \triangleq \underset{z\in\mathcal{R}^n}{\text{argmin}} \quad \left(g(z) + \tfrac{1}{2\tau}||z-x||^2\right), x \in \mathcal{R}^n, \tag{2}$$

is computed with complexity $\mathcal{O}(n)$ [2]. Herein $||\cdot||$ denotes the l_2 norm.

Problems that share the same structure as (1) arise quite often in different scientific disciplines, such as signal and image processing, data analysis, and machine learning. Typical applications in which the formulation given in (1) is relevant include compressive sensing, phase retrieval problems, medical imaging, dictionary learning, and many more (see [3,5–7,4] and references therein). When considering applications, the variable x represents the model parameters, whereas the role of $f(x)$ is to ensure a good fit between the observed data and the estimated parameters. In signal processing applications, $g(x)$ acts as a regularizer and typically takes the form of some parameter shrinkage norm, i.e., l_2 norm [8,9], sparsity-enforcing norm, i.e., l_1 norm [10–12], or its counterpart for the rank function, i.e., the nuclear norm [13,14]. Another popular structure for $g(x)$ is the Chebyshev norm, i.e., the l_∞ norm [15]. The function $g(x)$ can also be used to embed convex constraints, in which case it would act as an indicator function of some closed convex set [1].

In the context of large-scale optimization [16], problems that share the same structure as (1) are solved iteratively using different first-order optimization algorithms [17,18]. The bounds on the performance of black-box first-order methods have been es-

* Corresponding author.
E-mail addresses: endrit.dosti@aalto.fi (E. Dosti), sergiy.vorobyov@aalto.fi (S.A. Vorobyov), themistoklis.charalambous@aalto.fi (T. Charalambous).

https://doi.org/10.1016/j.sigpro.2022.108889
0165-1684/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

tablished by Nemirovsky and Yudin [19]. Loosely speaking, a first-order method is optimal in the black-box framework if it achieves the accelerated convergence rate with respect to the iteration counter k, while at the same time complying with the lower complexity bounds. The question of how to construct practical methods that are optimal has attracted the attention of the research community over decades. One of the first methods that managed to achieve the accelerated convergence rate in the black-box framework was the heavy ball method [20]. Therein, the acceleration is achieved by adding a momentum term to the gradient step, which nudges the new iterate in the direction of the previous step. The first method that is optimal in the sense of Nemirovsky and Yudin [19] is the Fast Gradient Method (FGM) [21]. It is built based on the mathematical machinery of estimating sequences, and has been since widely studied [22–27].

Finding different reasons behind acceleration has attracted significant attention in the recent research on first-order optimization. In [28], the authors have constructed accelerated first-order methods by exploiting the linear coupling between mirror and gradient descent. The framework presented therein leads to a myriad of applications wherein classical accelerated gradient methods do not apply, however all these applications are limited to the case of differentiable objective functions. The authors of [29] have derived an accelerated first-order method, which was inspired by the ellipsoid method. The proposed method is efficient; however, it suffers from the drawback that it requires an exact line search. An interesting framework is established in Flammarion and Bach [30], Su et al. [31], wherein the authors model the continuous-time limit of FGM as a second-order differential equation (ODE). Then, FGM equations can be obtained based on such a framework. Specifically, in Flammarion and Bach [30], the authors show that several accelerated schemes can be formulated as constant parameter ODE algorithms, wherein the stability of the systems would be equivalent to covergence at rate $\mathcal{O}(1/n^2)$. The limitation of the work is that the analysis presented therein is restricted only to the class of smooth and non-strongly convex problems. Moreover, in Su et al. [31] the authors show that the ODE type of analysis allows for a better understanding of Nesterov's scheme. However, the family of methods obtained therein, exhibits a similar convergence rate to FGM. Similar convergence rate as those obtained for FGM can also be derived by using theory from robust control [32]. A novel approach for analyzing the worst-case performance of first-order black-box methods has appeared in Drori and Teboulle [33]. The analysis conducted therein relies on the observation that the worst-case behavior improvement of a black-box method is itself an optimization problem, which is referred to as the performance estimation problem. By utilizing this approach, the authors of Kim and Fessler [34], 35] have introduced optimized first-order methods that are efficient and achieve a convergence bound that is two times smaller than the one attained by FGM. However, the development of these algorithms is restricted to solving problems with smooth objective functions.

Among the various approaches to the acceleration of first-order methods that were discussed above, the methods that were built based on the machinery of estimating sequences have attracted a lot of attention (see d'Aspremont et al. [18], Bubeck [36] and references therein). Several reasons that have led to their success are summarized in the sequel. First, on a theoretical level, FGM-type methods are proven to be optimal in the sense of Nemirovsky and Yudin [19]. Second, their practical performance is competitive even when they are used in conjunction with simple line search strategies, such as backtracking [37,38]. Third, they can be scaled to construct accelerated second-order methods [39,40] and accelerated higher-order methods [41,42]. Last, they have been shown to excel in performance even when they have been extended to other settings, such as distributed optimization [43,44], nonconvex optimization [45,46], stochastic optimization [47,48], non-Euclidean optimization [49,50], etc. In [51], it is argued that the key behind constructing optimal methods lies in the accumulation of some global information on the objective function. The mathematical objects which enable for capturing the relevant topological information on the function that is to be minimized are the estimating sequences. Typically, they consist of a pair of sequences, that simultaneously allow for parsing global information around the iterates, as well as for measuring the convergence rate of the minimization process. Despite their remarkable properties, estimating sequences exhibit the issue that there is no unique or systematic approach for constructing them. As we will see in the sequel, making the adequate choice of the estimating functions that comprise the estimating sequences can significantly impact the practical performance of the resulting algorithm.

The estimating sequences framework for the study and analysis of various methods has been presented in Baes [52]. An existing estimating sequence method that can directly solve (1) is the Accelerated Multistep Gradient Scheme (AMGS) [1]. The method is proven to enjoy the accelerated rate of convergence $\mathcal{O}(\frac{1}{k^2})$. Despite its notable theoretical and practical performance as measured by the number of iterations carried through until convergence, the method suffers the drawback that it requires two projection-like operations per iteration. This results in an increase of the computational burden, which (in the case of large-scale problems) is also reflected in an increase of the runtime of the method. This problem has been solved by the development of the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [53]. The method also enjoys the accelerated convergence rate of $\mathcal{O}(\frac{1}{k^2})$, while at the same time requiring only one projection-like operation per iteration. Similarly to Nesterov [21], FISTA does not explicitly utilize the machinery of estimating sequences. However, as has been demonstrated in Florea and Vorobyov [54], by properly selecting the estimating functions it is possible to establish links between FISTA and estimating sequence methods.

As discussed above, many of the existing seminal methods such as AMGS, FISTA and FGM [51, Constant Step Scheme I (2.2.19)], were obtained by explicitly (or implicitly) using the estimating sequences framework, and they all enjoy the theoretical accelerated rate of convergence. Despite being accelerated in theory, these methods still exhibit the following differences: *i*) The algorithmic structure of the methods changes depending on the different estimating sequences that are used in devising these algorithms. *ii)* The practical performance of the methods varies significantly when they are tested on real-world problems and datasets. Moreover, based on preliminary experiments that we have conducted for the cases of differentiable convex functions, we have observed that FGM converges faster than both AMGS and FISTA. Thus, the question of how to construct newer classes of estimating sequences that can be used to build more efficient methods for solving problems with composite objective structure arises. In this work, we answer this question affirmatively, and show that, by constructing the appropriate estimating functions, it is possible to devise very efficient accelerated first-order methods. More specifically, the main contributions of the article are as follows.

- In this work, we extend the existing estimating sequences framework presented in Nesterov [51] for minimizing differentiable objective functions, to the broader class of solving problems with composite structure given in (1).
- We introduce a new structure for the estimating functions, which we call the *composite estimating functions*. The proposed estimating functions are constructed by utilizing the gradient mapping technique [19] together with a tighter global lower bound on the objective function than the one obtained from the Taylor series expansion of a convex function.

- We show that our proposed estimating functions can be used to efficiently parse information around all the iterates, as well as measure the convergence rate of the minimization process. Unlike the estimating functions devised in Nesterov [51], which are only defined for the problem of minimizing smooth functions, our proposed composite estimating functions make use of the tighter lower bound on the objective function, as well as the subgradients of the objective function. This allows for designing methods that are used for solving a broader class of problems.
- We show how the proposed estimating sequences can be used to produce a new class of Composite Objective Multi-step Estimating-sequence Techniques (COMET), which are also endowed with an efficient line-search strategy. Unlike AMGS, the resulting algorithms require only one projection-like operation per iteration.
- We prove that COMET enjoys the accelerated convergence rate even when the Lipschitz constant is not known and needs to be estimated.
- We establish that the initialization of COMET can be made robust to the imperfect knowledge of the strong convexity parameter. Such a fact is very important for many practical applications, as computing the true value of the strong convexity parameter is computationally expensive.
- Through extensive simulations for various typical signal processing problems with composite structure, we show that the proposed method yields a better performance than the existing benchmarks. Furthermore, we also show the robustness of the selected instances of COMET with respect to the imperfect knowledge of the strong convexity parameter and the Lipschitz constant. To demonstrate the robustness, as well as the reliability of our proposed method, we test its performance on real-world datasets.

The article is organized as follows. In Section 2, we introduce the key assumptions of the paper, as well as some of the main concepts that are used in developing our method. In Section 3, we introduce the proposed estimating sequences for composite objectives and devise COMET based on them. In Section 4, we formally establish the convergence of COMET and derive the convergence rate for the minimization process. Then, in Section 5, we illustrate the performance of our proposed method in solving several optimization problems and show that it outperforms the existing benchmarks. Last, in Section 6, we present our conclusions and discuss possible future research directions.

2. Preliminaries

Assume that the objective function is bounded below, i.e., (1) has a solution. Another key assumption, which holds true for typical signal processing applications, is that the function and gradient computations have approximately the same complexity. For the problem setting of interest, the necessary oracle functions are the function evaluators, $f(x)$, $g(x)$, gradient evaluator $\nabla f(x)$, and proximal evaluator $\text{prox}_{\tau g}(x)$.

To simplify our analysis, let us relocate the strong convexity of $g(x)$ within the objective function in (1). Let $x_0 \in \mathcal{R}^n$ and consider that

$$F(x) = \left(f(x) + \frac{\tau\mu_g}{2}||x-x_0||^2\right) + \tau\left(g(x) - \frac{\mu_g}{2}||x-x_0||^2\right) = \hat{f}(x) + \tau\hat{g}(x). \tag{3}$$

The resulting function $\hat{f}(x)$ has a Lipschitz constant $L_{\hat{f}} = L_f + \tau\mu_g$ and strong convexity parameter $\mu_{\hat{f}} = \mu_f + \tau\mu_g$. On the other hand, the function $\hat{g}(x)$ has a strong convexity parameter $\mu_{\hat{g}} = 0$.

Recall that it is possible to construct upper and lower bounds for the smooth and strongly convex function $\hat{f}(x)$ by using the following relations:

$$\hat{f}(x) \le \hat{f}(y) + \nabla\hat{f}(y)^T(x-y) + \frac{L_{\hat{f}}}{2}||y-x||^2, \tag{4}$$

$$\hat{f}(x) \ge \hat{f}(y) + \nabla\hat{f}(y)^T(x-y) + \frac{\mu_{\hat{f}}}{2}||y-x||^2, \tag{5}$$

for all points $y \in \mathcal{R}^n$. Similarly, we can construct the following lower bound for the non-smooth term

$$\hat{g}(x) \ge \hat{g}(y) + s(y)^T(x-y), \tag{6}$$

where $s(y)$ is a subgradient of the function $\hat{g}(\cdot)$ at the point y. Moreover, for all $y \in \mathcal{R}^n$ and $L \ge L_{\hat{f}}$, we define

$$m_L(y;x) \triangleq \hat{f}(y) + \nabla\hat{f}(y)^T(x-y) + \frac{L}{2}||x-y||^2 + \tau\hat{g}(x). \tag{7}$$

Using the upper bound on the function established in (4), it can be seen that

$$m_L(y;x) \ge F(x), \forall x, y \in \mathcal{R}^n. \tag{8}$$

At this point, the composite gradient mapping can be introduced as

$$T_L(y) \triangleq \arg\min_{x\in\mathcal{R}^n} m_L(y;x). \tag{9}$$

Lastly, the composite reduced gradient can be defined as

$$r_L(y) \triangleq L(y - T_L(y)). \tag{10}$$

Let us now make a digression and note that when $\tau = 0$, we have the following: *i)* $\hat{f}(x) = f(x)$, which follows from (3); *ii)* $T_L(y) = y - \frac{\nabla\hat{f}(y)}{L}$, which follows from (9) and (7). Substituting these results into the definition given in (10), yields $r_L(y) = \nabla F(y) = \nabla f(y)$, i.e., the composite reduced gradient becomes the gradient of the objective function.

Returning back to the more general case, i.e., $\tau \neq 0$, from the first-order optimality conditions for (9), we can write

$$\nabla m_L(y; T_L(y))^T(x - T_L(y)) \ge 0,$$
$$(\nabla f(y) + L(T_L(y) - y) + \tau s_L(y))^T(x - T_L(y)) \ge 0, \tag{11}$$

where $s_L(y) \in \partial F(T_L(y))$ is a subgradient belonging to the subdifferential of $F(T_L(y))$, whose value depends on the point y. Equating the first bracket of (11) to 0, as well as recalling definition (10), we obtain the following relation, which is useful for computing the value of the composite reduced gradient

$$r_L(y) = L(y - T_L(y)) = \nabla f(y) + \tau s_L(y). \tag{12}$$

Last, we present a tighter lower bound on the objective function.

Theorem 1. *Let $F(x)$ be a composition of an $L_{\hat{f}}$-smooth and $\mu_{\hat{f}}$-strongly convex function $\hat{f}(x)$, and a simple convex function $\hat{g}(x)$, as given in (3). For $L \ge L_{\hat{f}}$, and $x, y \in \mathcal{R}^n$ we have*

$$F(x) \ge \hat{f}(T_L(y)) + \tau\hat{g}(T_L(y)) + r_L(y)^T(x-y) + \frac{\mu_{\hat{f}}}{2}||x-y||^2 + \frac{1}{2L}||r_L(y)||^2, \tag{13}$$

where $T_L(y)$ and $r_L(y)$ are defined in (9) and (10), respectively.

Proof. See Appendix A. □

3. COMET

In this section, we devise our proposed method. We start by introducing the composite estimating sequences, and then show why these sequences are useful. We also present a pair of composite estimating functions and show how to compute them recursively. Then, utilizing the proposed construction of the composite estimating functions, we derive COMET.

We begin by defining the composite estimating sequences.

Definition 1. The sequences $\{\phi_k\}_{k=0}^{\infty}$ and $\{\lambda_k\}_{k=0}^{\infty}$, $\lambda_k \geq 0$, are called composite estimating sequences of the function $F(\cdot)$ defined in (3), if $\lambda_k \to 0$ as $k \to \infty$, and $\forall x \in \mathcal{R}^n$, $\forall k \geq 0$ we have

$$\phi_k(x) \leq \lambda_k \phi_0(x) + (1-\lambda_k)F(x). \tag{14}$$

These composite estimating sequences allow for measuring the convergence rate to optimality, which is characterized in the following lemma.

Lemma 1. *If for some sequence of points $\{x_k\}_{k=0}^{\infty}$ we have $F(x_k) \leq \phi_k^* \triangleq \min_{x\in\mathcal{R}^n} \phi_k(x)$, then $F(x_k) - F(x^*) \leq \lambda_k[\phi_0(x^*) - F(x^*)]$, where $x^* = \arg\min_{x\in\mathcal{R}^n} F(x)$.*

Proof. See Appendix B. □

We are now ready to show how the composite estimating sequences can be defined recursively.

Lemma 2. *Assume that there exists a sequence $\{\alpha_k\}_{k=0}^{\infty}$, where $\alpha_k \in (0,1)$ $\forall k$, such that $\sum_{k=0}^{\infty}\alpha_k = \infty$, and an arbitrary sequence $\{y_k\}_{k=0}^{\infty}$. Furthermore, let $\lambda_0 = 1$ and assume that the estimates L_k of the Lipschitz constant $L_{\hat{f}}$ are selected in a way that inequality (4) is satisfied for all the iterates x_k and y_k. Then, the sequences $\{\phi_k\}_{k=0}^{\infty}$ and $\{\lambda_k\}_{k=0}^{\infty}$, which are defined recursively as*

$$\lambda_{k+1} = (1-\alpha_k)\lambda_k, \tag{15}$$

$$\begin{aligned}\phi_{k+1}(x) = {} & (1-\alpha_k)\phi_k(x) + \alpha_k\Big(F\big(T_{L_k}(y_k)\big) + \frac{1}{2L_k}\|r_{L_k}(y_k)\|^2\Big) \\ & + \alpha_k\Big(r_{L_k}(y_k)^T(x-y_k) + \frac{\mu_f}{2}\|x-y_k\|^2\Big),\end{aligned} \tag{16}$$

are composite estimating sequences.

Proof. See Appendix C. □

At this point, we provide a comparison between the results obtained in Lemmas 1 and 2 to their counterpart devised for the simpler case of minimizing smooth convex functions presented in Nesterov [51]. First, we can see from Lemma 1 that the convergence rate of the minimization process depends entirely on the rate at which $\lambda_k \to 0$. Moreover, the result hints that for problem (1) we should expect a similar convergence rate as in the simpler case of minimizing a differentiable convex function. Then, in Lemma 2, we have shown how to form the estimating functions. It can also be seen from (16) that we are utilizing a tighter lower bound than the one used for deriving FGM for the smooth strongly convex case.[1] Furthermore, it can be noted that the cost function is evaluated at specific points in its domain, which are produced by the composite gradient mapping. Last, it can be observed that the subgradient of the non-smooth objective function is needed to construct the estimating functions $\{\phi_k\}_{k=0}^{\infty}$.

Until now, no particular structure for the functions in the sequence $\{\phi_k\}_{k=0}^{\infty}$ has been proposed yet. Inspired by the analysis for FGM in the setup of smooth convex functions [51], in the sequel we let

$$\phi_k(x) \triangleq \phi_k^* + \frac{\gamma_k}{2}||x-v_k||^2, \ \forall k = 1,2,\ldots, \tag{17}$$

where $\gamma_k \in \mathcal{R}^+$ and $v_k \in \mathcal{R}^n$, $\forall k = 0,1,\ldots$. Nevertheless, we stress that this selection is not unique. As a matter of fact, different choices of the canonical structure for the function $\phi_k(x)$ can lead to entirely different algorithms, see for example [49,56,55]. Next, in Lemma 3 we show how the terms $\{\gamma_k\}_{k=0}^{\infty}$, $\{v_k\}_{k=0}^{\infty}$ and $\{\phi_k^*\}_{k=0}^{\infty}$ can be computed recursively.

Lemma 3. *Let $\phi_0(x) = \phi_0^* + \frac{\gamma_0}{2}||x-v_0||^2$, where $\gamma_0 \in \mathcal{R}^+$ and $v_0 \in \mathcal{R}^n$. Then, the process defined in Lemma 2 preserves the canonical form of the function $\{\phi_k(x)\}_{k=0}^{\infty}$ presented in (17), where the sequences $\{\gamma_k\}_{k=0}^{\infty}$, $\{v_k\}_{k=0}^{\infty}$ and $\{\phi_k^*\}_{k=0}^{\infty}$ can be computed as follows*

$$\gamma_{k+1} = (1-\alpha_k)\gamma_k + \alpha_k\mu_{\hat{f}}, \tag{18}$$

$$v_{k+1} = \frac{1}{\gamma_{k+1}}\Big((1-\alpha_k)\gamma_k v_k + \alpha_k\big(\mu_{\hat{f}}y_k - L_k\big(y_k - T_{L_k}(y_k)\big)\big)\Big), \tag{19}$$

$$\begin{aligned}\phi_{k+1}^* = {} & (1-\alpha_k)\phi_k^* + \alpha_k\Big(F\big(T_{L_k}(y_k)\big) + \frac{1}{2L_k}||r_{L_k}(y_k)||^2\Big) \\ & - \frac{L_k^2\alpha_k^2}{2\gamma_{k+1}}||y_k - T_{L_k}(y_k)||^2 + \frac{\mu_{\hat{f}}\alpha_k\gamma_k(1-\alpha_k)}{2\gamma_{k+1}}||y_k - v_k||^2 \\ & + \frac{L_k\alpha_k\gamma_k(1-\alpha_k)}{\gamma_{k+1}}(y_k - v_k)^T(y_k - T_{L_k}(x_k)).\end{aligned} \tag{20}$$

Proof. See Appendix D. □

Comparing the result obtained in Lemma 3 with its counterpart constructed for minimizing smooth objective functions [51, Lemma 2.2.3], it can be seen that the recursion for computing the elements in the sequences $\{v_k\}_{k=0}^{\infty}$ and $\{\phi_k^*\}_{k=0}^{\infty}$ has changed. It now reflects both the different lower bound on the objective function, as well as the reduced composite gradient, which were utilized for constructing the composite estimating functions.

Let us now proceed to constructing the algorithm via induction. First, let $\phi_0^* = F(x_0)$. Next, assume that for some iteration k, we have: $\phi_k^* \geq F(x_k)$. To conclude the induction argument, we need to show that $\phi_{k+1}^* \geq F(x_{k+1})$. Using the aforementioned assumption for iteration k into (20), it can be written that

$$\begin{aligned}\phi_{k+1}^* \geq {} & (1-\alpha_k)F(x_k) + \alpha_k\Big(F\big(T_{L_k}(y_k)\big) + \frac{1}{2L_k}||r_{L_k}(y_k)||^2\Big) \\ & - \frac{{L_k}^2\alpha_k^2}{2\gamma_{k+1}}||y_k - T_{L_k}(y_k)||^2 + \frac{\mu_{\hat{f}}\alpha_k\gamma_k(1-\alpha_k)}{2\gamma_{k+1}}||y_k - v_k||^2 \\ & + \frac{L_k\alpha_k\gamma_k(1-\alpha_k)}{\gamma_{k+1}}(v_k - y_k)^T(y_k - T_{L_k}(y_k)).\end{aligned} \tag{21}$$

Then, substituting the bound obtained in Theorem 1, as well as (10) into (21), we obtain

$$\begin{aligned}\phi_{k+1}^* \geq {} & (1-\alpha_k)\Big(F(T_{L_k}(y_k)) + r_{L_k}(y_k)^T(x_k - y_k) + \frac{\mu}{2}||x_k - y_k||^2 \\ & + \frac{1}{2L_k}||r_{L_k}(y_k)||^2\Big) + \alpha_k\Big(F\big(T_{L_k}(y_k)\big) + \frac{1}{2L_k}||r_{L_k}(y_k)||^2\Big) \\ & - \frac{\alpha_k^2}{2\gamma_{k+1}}||r_{L_k}(y_k)||^2 + \frac{\mu\alpha_k\gamma_k(1-\alpha_k}{2\gamma_{k+1}}||y_k - v_k||^2 \\ & + \frac{\alpha_k\gamma_k(1-\alpha_k)}{\gamma_{k+1}}r_{L_k}(y_k)^T(v_k - y_k).\end{aligned} \tag{22}$$

Making some algebraic manipulations and factoring in (23), we reach

[1] Recall that when $F(x)$ is smooth and convex function, the composite reduced gradient becomes just the gradient of the function.

$$\phi^*_{k+1} \geq F(T_{L_k}(y_k)) + \left(\frac{1}{2L_k} - \frac{\alpha_k^2}{2\gamma_{k+1}}\right)||r_{L_k}(y_k)||^2 + (1-\alpha_k) r_{L_k}(y_k)^T \left(x_k - y_k + \frac{\alpha_k \gamma_k}{\gamma_{k+1}}(v_k - y_k)\right). \tag{23}$$

At this point, a relation for the unknown terms in the sequences $\{\alpha_k\}_{k=0}^{\infty}$ and $\{y\}_{k=0}^{\infty}$ needs to be found. Observe that in (24) we can obtain the update rule for the terms in the sequence $\{\alpha_k\}_{k=0}^{\infty}$ as

$$\alpha_k = \sqrt{\frac{\gamma_{k+1}}{L_k}}. \tag{24}$$

Utilizing the recursion for γ_{k+1} given by (18), and solving the resulting quadratic equation yields

$$\alpha_k = \frac{\mu_{\hat{f}} - \gamma_k + \sqrt{\left(\mu_{\hat{f}} - \gamma_k\right)^2 + 4L_k\gamma_k}}{2L_k}. \tag{25}$$

Making the aforementioned selection for α_k, (24) can now be written as

$$\phi^*_{k+1} \geq F(T_{L_k}(y_k)) + (1-\alpha_k) r_{L_k}(y_k)^T \left(x_k - y_k + \tfrac{\alpha_k\gamma_k}{\gamma_{k+1}}(v_k - y_k)\right). \tag{26}$$

Thus, the update rule for the term y_k can be obtained by setting

$$x_k - y_k + \frac{\alpha_k\gamma_k}{\gamma_{k+1}}(v_k - y_k) = 0. \tag{27}$$

This results in

$$y_k = \frac{\gamma_{k+1}x_k + \alpha_k\gamma_k v_k}{\gamma_{k+1} + \alpha_k\gamma_k}. \tag{28}$$

To establish that $\phi_{k+1} \geq F(x_{k+1})$, it suffices to let $x_{k+1} = T_{L_k}(y_k)$.

Last, another major difference between our proposed method and its counterpart for minimizing differentiable convex functions [51], is the fact that our analysis allows for the line search adaptation.[2] The goal of our proposed line-search strategy is to select the smallest constant L_k such that (4) is satisfied $\forall k = 0, 1, \ldots$. To progress faster towards x^* in the initial iterations, we would want to initialize $L_0 \in]0, L_{\hat{f}}[$, and then gradually increase the value of the estimate of the Lipschitz constant across the iterations. However, since the true value of $L_{\hat{f}}$ is not known, this approach cannot be used. Therefore, it would be more preferable to select the line search strategy such that it ensures the robustness of the method with respect to the initialization of the estimate of the Lipschitz constant and ensure a dynamic update of the step size. Such a scheme would be of importance for many applications in signal processing (see Florea and Vorobyov [54] and the references therein). For this purpose, the following two parameters can be utilized: *i)* a constant $\eta_u > 1$, which increases the value of the estimate; *ii)* a constant $\eta_d \in]0, 1[$, which decreases the value of the estimate of the Lipschitz constant. Finally, the proposed method is summarized in Algorithm 1.

Comparing between our proposed method and FGM (Constant Step Scheme I in Nesterov [51]), we can observe from lines 6 and 7 in Algorithm 1, the similarities in updating the sequences $\{\alpha_k\}_{k=0}^{\infty}$ and $\{\gamma_k\}_{k=0}^{\infty}$. A difference can, however, be noticed in the update of the terms in the sequence $\{y_k\}_{k=0}^{\infty}$, whose value becomes independent of $\mu_{\hat{f}}$. Additionally, a key difference between the methods is in the update of the iterates x_k. Due to the composite structure of the objective function of interest, the next iterate x_{k+1} is computed by taking a proximal gradient step. Note that as long as the non-smooth term $g(x)$ has a simple structure, the proximal term

[2] Note that several backtracking strategies have already been proposed in the literature (see for example Nesterov [1], Beck and Teboulle [53], Tseng [57]).

Algorithm 1 COMET.

Input: $x_0 \in \mathcal{R}^n$, $L_0 > 0$, $\mu_{\hat{f}}$, $\gamma_0 \in [0, 3L_0 + \mu_{\hat{f}}]$, $\eta_u > 1$ and $\eta_d \in]0, 1[$.

1: **while** $k \leq K_{\max}$ **do**
2: $\quad \hat{L}_i \leftarrow \eta_d L_k$
3: $\quad$ **while** True **do**
4: $\quad\quad \hat{\alpha}_i \leftarrow \frac{(\mu_{\hat{f}} - \gamma_k) + \sqrt{(\mu_{\hat{f}} - \gamma_k)^2 + 4\hat{L}_i\gamma_k}}{2\hat{L}_i}$
5: $\quad\quad \hat{\gamma}_{i+1} \leftarrow (1 - \hat{\alpha}_i)\gamma_k + \hat{\alpha}_i\mu_{\hat{f}}$
6: $\quad\quad \hat{y}_i \leftarrow \frac{\hat{\gamma}_{i+1}x_k + \hat{\alpha}_i\gamma_k v_k}{\hat{\gamma}_{i+1} + \hat{\alpha}_i\gamma_k}$
7: $\quad\quad \hat{x}_{i+1} \leftarrow \text{prox}_{\frac{1}{\hat{L}_i}\hat{g}}\left(\hat{y}_i - \frac{1}{\hat{L}_i}\nabla f(\hat{y}_i)\right)$
8: $\quad\quad \hat{v}_{i+1} \leftarrow \frac{1}{\hat{\gamma}_{i+1}}\left((1 - \hat{\alpha}_i)\gamma_k v_k + \hat{\alpha}_i\left(\mu_{\hat{f}}\hat{y}_i - \hat{L}_i(\hat{y}_i - \hat{x}_{i+1})\right)\right)$
9: $\quad\quad$ **if** $F(\hat{x}_{i+1}) \leq m_{\hat{L}_i}(\hat{y}_i, \hat{x}_{i+1})$ **then**
10: $\quad\quad\quad$ Break from loop
11: $\quad\quad$ **else**
12: $\quad\quad\quad \hat{L}_{i+1} \leftarrow \eta_u\hat{L}_i$
13: $\quad\quad$ **end if**
14: $\quad\quad i \leftarrow i + 1$
15: $\quad$ **end while**
16: $\quad L_{k+1} \leftarrow \hat{L}_i$, $x_{k+1} \leftarrow \hat{x}_i$, $\alpha_k \leftarrow \hat{\alpha}_{i-1}$, $y_k \leftarrow \hat{y}_{i-1}$, $i \leftarrow 0$, $k \leftarrow k + 1$
17: **end while**

Output: x_k

can be computed efficiently. Another major difference between the methods lies in the update of the terms in the sequence $\{v_k\}_{k=0}^{\infty}$, which now reflect the usage of the proposed subgradient. Last, the parameter γ_0 can now be selected over a wider range of parameters than what is guaranteed by the existing convergence results for FGM established in Nesterov [51, Lemma 2.2.4]. The rationale behind this result will become clear in the sequel.

Before we proceed to analyzing the convergence rate of the minimization process, let us evaluate the behavior of the estimate of the Lipschitz constant. Depending on the initialization of L_0, there are two scenarios.

i) If $L_0 \in]0, L_{\hat{f}}[$, then from line 11 in Algorithm 1, it can be observed that the estimate of the Lipschitz constant at iteration k increases only if $L_{k-1} \leq L_{\hat{f}}$. Therefore, we can write

$$L_0 \leq \hat{L}_i \leq L_k \leq \eta_u L_{\hat{f}}. \tag{29}$$

ii) If $L_0 \geq L_{\hat{f}}$, then the condition in line 11 of Algorithm 1 is satisfied, and estimate of the Lipschitz constant cannot increase further. This yields

$$L_k \leq \eta_d L_0. \tag{30}$$

Combining the bounds (30) and (31), we can see that despite the initialization of L_0, it is always true that

$$L_k \leq L_{\max} \triangleq \max\{\eta_d L_0, \eta_u L_{\hat{f}}\}. \tag{31}$$

To obtain an easier understanding of the proposed method, we also present the flowchart in Fig. 1. As can be seen from the flowchart, at any iteration k the inputs are feed into the outer loop, which starts by decreasing the estimate of the Lipschitz constant (see line 2 in Algorithm 1). The inner loop then updates the parameters and takes *one* proximal gradient step to produce the iterate at iteration $k+1$ (see lines 4–8 in Algorithm 1). As long as a function-based stopping criterion is not satisfied, the inner loop also corrects the value of the estimate of the Lipschitz contant, which corresponds to line 12 in Algorithm 1. After the

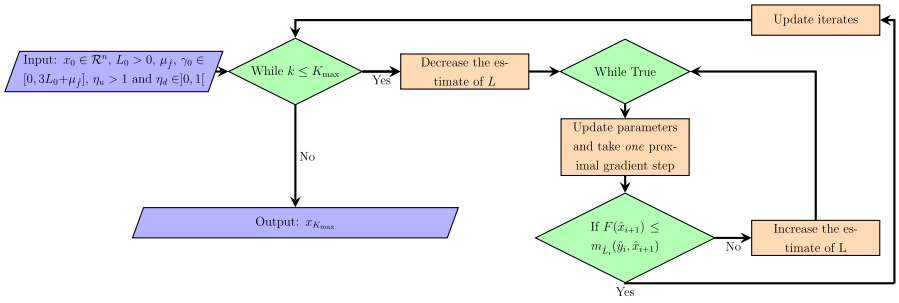

Fig. 1. Flowchart that depicts the main building blocks of our proposed method.

function-based stopping criterion is satisfied, the inner loop is terminated and the method proceeds to the next iterate (see line 16 in Algorithm 1). The numerical procedure terminates after the iteration-based stopping criterion is satisfied, and outputs $x_{K_{\max}}$. Contrasting our proposed COMET to AMGS and FISTA we can highlight several differences. First, with respect to AMGS, we note that the methods require different input parameters. Moreover, observe that our proposed COMET only queries one proximal and one gradient oracle to update the iterates. On the other hand, AMGS requires double the queries. As we will see in Section 5.3, this translates into an increase in the runtime of AMGS. Comparing our proposed COMET to FISTA, we note that they both query a single proximal and gradient oracle to update the iterates. The first difference in the methods lies in the line-search procedure that is employed by COMET, which is more efficient as it allows for dynamically updating the estimate of the Lipschitz constant. On the other hand, the line-search procedure proposed for FISTA only allows for increasing the estimate of the Lipschitz constant. Another major difference between the methods lies in the fact that the methods are initialized using different input parameters. Similar to the differences with AMGS, this arises because the methods were devised using different principles of acceleration of first-order methods.

4. Convergence analysis

Let us begin by noting that the result obtained in Lemma 1 suggests that the convergence rate of the minimization process will be the same as the rate at which $\lambda_k \to 0$. This is made more precise in the following theorem.

Theorem 2. *If we let $\lambda_0 = 1$ and $\lambda_k = \prod_{i=0}^{k-1}(1-\alpha_i)$, Algorithm 1 generates a sequence of points $\{x_k\}_{k=0}^{\infty}$ such that*

$$F(x_k) - F(x^*) \leq \lambda_k \left[F(x_0) - F(x^*) + \frac{\gamma_0}{2}||x_0 - x^*||^2 \right]. \tag{32}$$

Proof. See Appendix E. □

Now, recall that from Definition 1, we must have $\lambda_k \to 0$. Therefore, the result of Theorem 2 is sufficient to establish the fact that the sequence of iterates produced by our proposed algorithm converges to the optimal solution. The next step is to evaluate the rate of convergence of this process. Let us begin by characterizing the rate at which $\lambda_k \to 0$.

Lemma 4. *For all $k \geq 0$, Algorithm 1 guarantees that*

1. *If $\gamma_0 \in [0, \mu_{\hat{f}}[$, then*

$$\lambda_k \leq \frac{2\mu_{\hat{f}}}{L_k\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}} - e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2} \leq \frac{2}{(k+1)^2}. \tag{33}$$

2. *If $\gamma_0 \in [\mu_{\hat{f}}, 3L_0 + \mu_{\hat{f}}]$, then*

$$\lambda_k \leq \frac{4\mu_{\hat{f}}}{(\gamma_0 - \mu_{\hat{f}})\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}} - e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2} \leq \frac{4L_k}{(\gamma_0 - \mu_{\hat{f}})(k+1)^2}. \tag{34}$$

Proof. See Appendix F. □

Comparing the results obtained in Lemma 4 with the earlier results obtained in Nesterov [51, Lemma 2.2.4], we can see two major differences. First, our proposed analysis establishes the convergence of the method even when the true value of the Lipschitz constant is not known. Second, we can see that it is possible to establish the convergence of the method in minimizing objective functions with composite structure for a wider initialization range of the parameter γ_0. The importance of this result arises from the fact that the method exhibits a faster theoretical and practical convergence when $\gamma_0 = 0$, which is not supported by the existing analysis for FGM. At the same time, the initialization $\gamma_0 = 0$ also provides robustness with respect to the imperfect knowledge of $\mu_{\hat{f}}$.

From Theorem 2, we can see that the convergence rate of the minimization process depends on the distance $F(x_0) - F(x^*)$. The following lemma yields an upper bound on it.

Lemma 5. *Let $F(x)$ be a convex function with composite structure as shown in (1). Moreover, let $T_L(y)$ and $r_L(y)$ be computed as given in (9) and (12), respectively. Then, for any starting point x_0 in the domain of $F(x)$, we have*

$$F(x_0) - F(x^*) \leq \frac{L_0}{2}||x_0 - x^*||^2. \tag{35}$$

Proof. See Appendix G. □

Combining the results of Lemmas 4 and 5 with Theorem 2, we can immediately obtain the convergence rate for COMET as follows.

Theorem 3. *Algorithm 1 generates a sequence of points such that*

1. *If $\gamma_0 \in [0, \mu_{\hat{f}}[$, then*

$$F(x_k) - F(x^*) \leq \frac{\mu_{\hat{f}}(L_0 + \gamma_0)||x_0 - x^*||^2}{L_k\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}} - e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2} \tag{36}$$

2. *If $\gamma_0 \in [\mu_{\hat{f}}, 3L_0 + \mu_{\hat{f}}]$, then*

$$F(x_k) - F(x^*) \leq \frac{2\mu_{\hat{f}}(L_0 + \gamma_0)||x_0 - x^*||^2}{(\gamma_0 - \mu_{\hat{f}})\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu}{L_k}}} - e^{-\frac{k+1}{2}\sqrt{\frac{\mu}{L_k}}}\right)^2} \tag{37}$$

From the result of Theorem 3 we can see that our proposed method is guaranteed to converge over a wider interval than its counterpart designed for minimizing smooth and strongly convex objectives. Notice that initializing $\gamma_0 = 0$ would guarantee the fastest convergence of the method. Such a result is important when considering many practical applications, wherein the true values of $\mu_{\hat{f}}$ and $L_{\hat{f}}$ are often not known and should be estimated. Another factor that impacts the rate of convergence of the minimization process is also the initialization of L_0. From (37), (38) we can see that the smaller the value of L_0, the faster the convergence of the method.

5. Numerical study

In this section, we compare the numerical performance of the proposed method against the two seminal black-box methods, namely, AMGS and FISTA, in solving several optimization problems, which arise often in many signal and image processing, statistics and data science applications. The selected loss functions are the quadratic and logistic loss functions, both with elastic net regularization. Moreover, we also test the performance of our proposed COMET in solving the regularized image deblurring problem. As we will see in the sequel, controlling the parameters of the elastic net regularizer allows for simulating extremely ill-conditioned examples. For the constructed examples, we show that COMET outperforms the selected benchmarks in terms of minimizing the number of iterations needed to achieve a certain tolerance level. To provide reliable results, we utilize both synthetic and real data, that are selected from the Library for Support Vector Machines [58]. To find the optimal solutions, we use CVX [59].

In the first example, we illustrate the performance of three variants of COMET: *1)* we consider the variant that in theory is expected to result in the fastest convergence, which is obtained when we initialize for $\gamma_0 = 0$, and it is referred to as "COMET, variant 1"; *2)* we also consider the variant that is expected to produce the slowest convergence, which happens when we initialize $\gamma_0 = 3L_0 + \mu_{\hat{f}}$, and it is labeled as "COMET, variant 2"; *3)* we also implement the variant of COMET that is obtained when $\gamma_0 = \mu_{\hat{f}}$, which is referred to as "COMET, variant 3". When comparing the performance of the methods under the condition where the Lipschitz constant is not known, for both AMGS and FISTA we utilize the line-search strategies presented in the respective works [1,53]. We note that throughout all the simulations the starting point x_0 is randomly selected and all algorithms are initialized in it. The numerical experiments are conducted using an Intel(R) Core(TM) i7-8665U 1.90 GHz CPU and the methods are implemented using Matlab.

5.1. Minimizing the quadratic loss function

Consider one of the most popular problems in signal processing and statistics

$$\underset{x\in\mathcal{R}^n}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{m}\left(a_i^T x - y_i\right)^2 + \frac{\tau_1}{2}\|x\|^2 + \tau_2\|x\|_1, \tag{38}$$

where $||\cdot||_1$ denotes the l_1 norm. The objective is to show that the theoretical gains of COMET, which are discussed in Section 4, are also reflected in the practical performance of the methods. Moreover, we analyze how the performance of the methods scales with the condition number of the problem. We also illustrate the practical benefits of utilizing the proposed line-search strategy.

Let us first consider the simplest case, where the Lipschitz constant is assumed to be known. It allows for an objective assessment of the effectiveness of the methods in finding the optimal solution. For this example, we utilize synthetic data. We consider the diagonal matrix $A \in \mathcal{R}^{m\times m}$ and sample the elements a_{ii} from the discrete set $\{10^0, 10^{-1}, 10^{-2}, \ldots, 10^{-\xi}\}$ uniformly at random. This choice of selecting A ensures that $L = 1$ and $\mu_f = 10^{-\xi}$, which results in the condition number 10^{ξ}. Then, we select the elements of the vector $y \in \mathcal{R}^m$ by uniformly drawing them from the box $[0, 1]^n$. Lastly, we note that in our computational experiments we set $m \in \{500, 1000, 1500, 2000\}$, $\xi \in \{3, 4, 7, 8\}$ and $\tau_1 = \tau_2 \in \{10^{-3}, 10^{-4}, 10^{-7}, 10^{-8}\}$.

From Fig. 2, we can observe that the proposed method significantly outperforms all the existing benchmarks. First, notice that the larger the condition number of the problems becomes, the more iterations, and consequently computations, are required by the methods to obtain a good solution. Comparing between the methods, we can observe that all instances of COMET yield a better quality of the obtained solution, as measured by the distance to x^*. Moreover, we can clearly see that the iterates produced by COMET converge to x^* in a much smaller number of iterations. Another important observation that can be made from the figure is that the proposed method exhibits better monotonic properties than both AMGS and FISTA. Comparing the performance of different variants of COMET, we can observe that their behavior is similar and the differences in performance are not too large. We can see that the variant that yields the best performance is the one obtained when $\gamma_0 = 0$, which is coherent with the theoretical results established in Section 4.

Next, we proceed to analyzing a more realistic scenario. We assume that the Lipschitz constant is not known, and needs to be estimated by using a line-search procedure. To demonstrate the robustness of the line-search strategy to be utilized in conjunction with COMET, we consider the following cases. *i)* The Lipschitz constant is underestimated by a factor of 10, i.e., $L_0 = 0.1L_{\hat{f}}$. *ii)* The Lipschitz constant is overestimated by a factor of 10, i.e., $L_0 = 10L_{\hat{f}}$. Moreover, we note that we selected $\eta_u = 2$ and $\eta_d = 0.9$, which were suggested in Becker et al. [60] because they ensure a good performance of the methods in many applications. Another parameter that is computationally expensive to be estimated in practice is the strong convexity parameter $\mu_{\hat{f}}$. To avoid an increase in computations, in all the following simulations we equate the value of the strong convexity parameter to that of the regularization term in the objective function in (41). Lastly, we note that for all the examples that will be shown in the sequel, we utilize the datasets "a1a" and "colon-cancer". The former dataset has data matrix $A \in \mathcal{R}^{1605\times 123}$, whereas the latter has $A \in \mathcal{R}^{62\times 2000}$.

For the datasets that we are utilizing, the respective Lipschitz constants are $L_{\text{"a1a'prime}} = 10061$ and $L_{\text{"colon-cancer'prime}} = 1927.4$. Moreover, we let the regularizer term $\tau_1 = \tau_2 \in \{10^{-5}, 10^{-6}\}$. Evidently, this selection of the regularizer terms guarantees a very large condition number $\kappa = \frac{L_{\hat{f}}}{\mu_{\hat{f}}}$ for the problems that are being solved. The numerical results are presented in Fig. 3, from which we can observe that all the instances of COMET significantly outperform the existing benchmarks. First, the final iterate produced by the first variant of COMET is the closest to x^*. This is most visible from the numerical experiments conducted on the "a1a" dataset, which are depicted in Fig. 3(a) and (b). Second, the iterates produced by the proposed COMET converge to x^* by requiring a significantly smaller number of iterations, when compared to AMGS and FISTA. Third, the performance of FISTA largely depends on the initialization of the Lipschitz constant. On the other hand, we can observe that for both datasets, the performance of both AMGS and COMET remains unaffected by the value of L_0. We stress that COMET retains the robustness to L_0 at the lower computational cost of only one projection-like operation per iteration, whereas AMGS requires double of that. Last, comparing the performance between the selected variants of COMET, we can see that in practice their performance differences are minor. Neverthe-

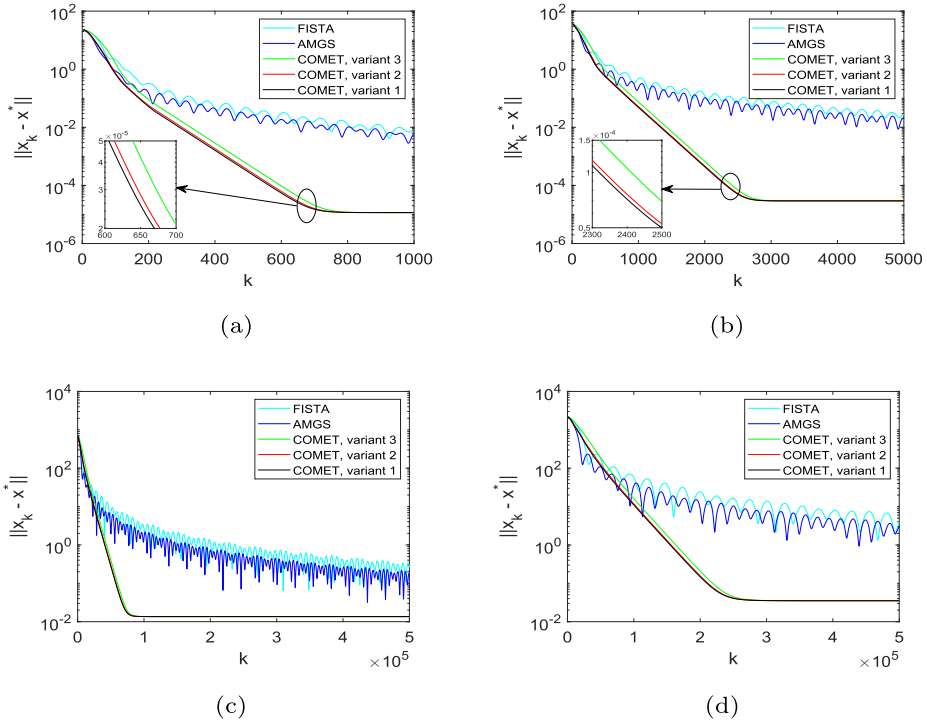

Fig. 2. Comparison between the efficiency of the algorithms tested in minimizing the quadratic loss function with elastic net regularizer on randomly generated data.

less, our results shown in Fig. 3(a) and (b) suggest that the version of COMET which is obtained when $\gamma_0 = 0$ yields a better performance. This becomes important particularly when considering practical applications, wherein the true values of $\mu_{\hat{f}}$ and $L_{\hat{f}}$ are typically not known and their true values can only be estimated within some error bounds. From this perspective, we can conclude that the instance of COMET obtained by setting $\gamma_0 = 0$ enjoys both the faster convergence of the iterates and the robustness with respect to the imperfect knowledge of $\mu_{\hat{f}}$ and $L_{\hat{f}}$.

5.2. Minimizing the logistic loss function

To demonstrate the versatility of the proposed black-box method, let us now compare its performance to the selected benchmarks in minimizing a regularized logistic loss function with elastic net regularizer

$$\underset{x\in\mathcal{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{i=1}^{m}\log\left(1+\mathrm{e}^{-b_i x a_i}\right)+\frac{\tau_1}{2}\|x\|^2+\tau_2\|x\|_1. \tag{39}$$

For this problem type, we diversify the utilized datasets and select "triazine", as well as a subset of "rcv1.binary". For the chosen datasets, we have $A_{\text{''triazine'}prime} \in \mathcal{R}^{186\times 61}$ and $A_{\text{''rcv1.binary'}prime} \in \mathcal{R}^{1000\times 2000}$. Moreover, from the results of Fig. 3, we have observed that the performance of FISTA has been dependent on the initial estimate of the Lipschitz constant and has been overall worsened when $L_{\hat{f}}$ is unknown. Therefore, to provide the fairest comparison with respect to FISTA, for this set of examples we estimate the value of L directly from the data. More specifically, we have $L_{\text{''triazine'}prime} = 25.15$ and $L_{\text{''rcv1.binary'}prime} = 1.13$. On the other hand, similar to the earlier computational experiments, we equate the value of the strong convexity parameter to that of the regularization term in the objective function in (40). Last, we note that for this set of numerical experiments we consider the cases when $\tau_1 \neq \tau_2$. The results are reported in Fig. 4, wherein the specific values for τ_1 and τ_2 are also presented.

From Fig. 4, we can observe that for both datasets, COMET outperforms and exhibits better monotonic properties than AMGS or FISTA. Moreover, all variants of COMET require a much lower number of iterations to produce iterates which are closest to x^*. Last, for the selected problem type, the variant of COMET which is constructed when $\gamma_0 = 0$ yields the best practical performance, although the true value of $\mu_{\hat{f}}$ is not known.

5.3. Application to the regularized image deblurring problem

Let us now consider solving the problem of regularized image deblurring, which we formulate as follows

$$\underset{x\in\mathcal{R}^n}{\text{minimize}} \quad \|RWx-y\|^2+\frac{\tau_1}{2}\|x\|^2+\tau_2\|x\|_1, \tag{40}$$

where R represents the blur operator and W is the inverse three-stage Haar wavelet transform. In this example, $x \in \mathcal{R}^{256\times 256}$ is the cameraman test image [53]. To blurr the image, we scale its pixels in the range [0,1], add zero-mean Gaussian noise with standard deviation 10^{-3} and apply the blur operator R. Moreover, we set the regularizer parameters $\tau_1 = 1\times\ 10^{-3}$ and $\tau_2 = 10^{-5}$. For this problem, we initialize $L_0 = L_F$, which is obtained as the maximum eigenvalue of $(RW)^T(RW)$, and set $\mu_F = \tau_1$. Different from the previous sections, herein we report the CPU runtime (in seconds) that was needed to decrease the value of the objective function. For a more extensive comparison, herein we have also included the Accelerated Composite Gradient Method (ACGM) [37], which is built on top of the estimating sequences variant that was used for designing AMGS. Moreover, we have also included the variant of FISTA presented in Chambolle and Pock [61], which is designed to exploit the strong convexity information that might be available about the objective function.

Our findings are summarized in Table 1. The first column was obtained by computing the values of the objective function that

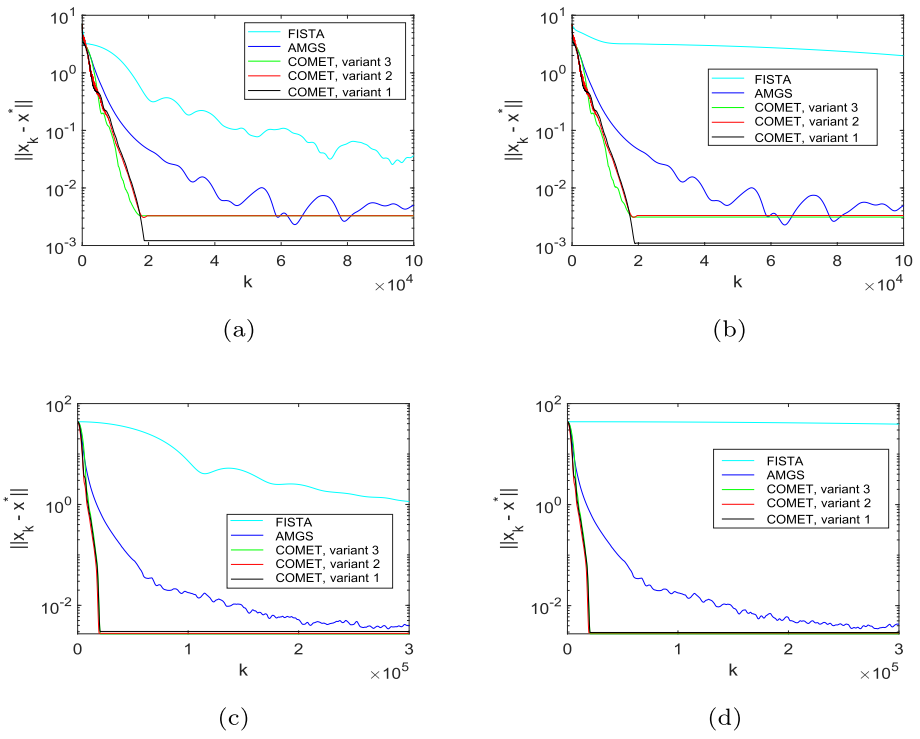

Fig. 3. Comparison between the efficiency and robustness with respect to the initialization of the Lipschitz constant of the algorithms tested in minimizing the quadratic loss function with elastic net regularizer on real data.

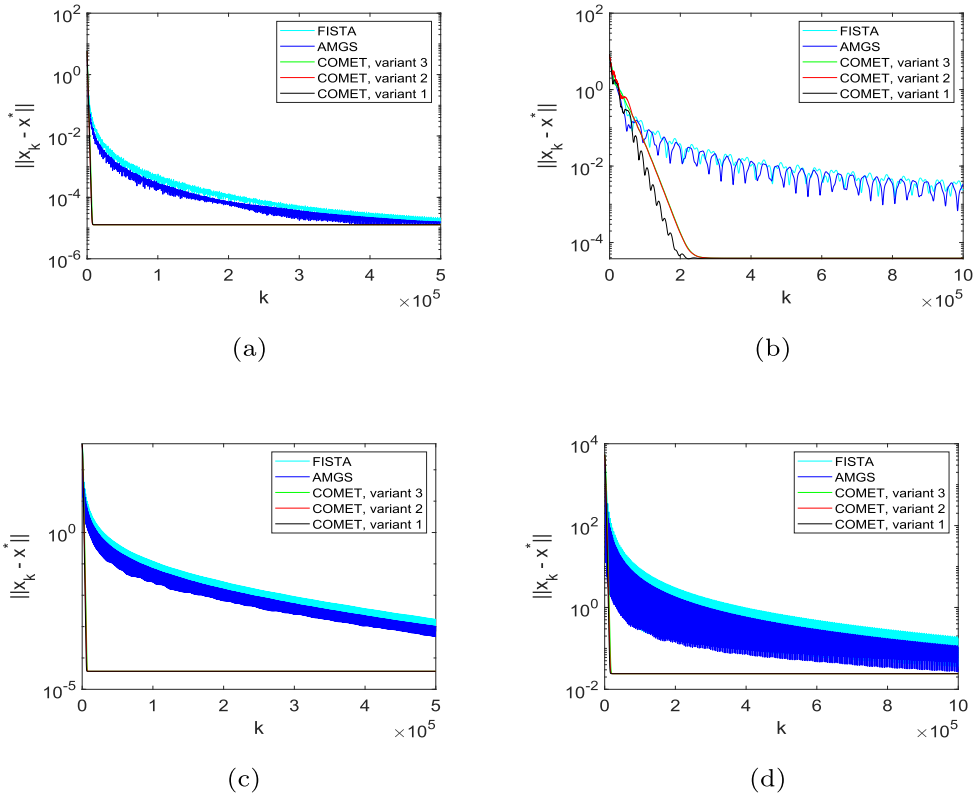

Fig. 4. Comparison between the efficiency of the algorithms tested in minimizing the logistic loss function with elastic net regularizer on real data.

Table 1
Comparison between the CPU runtimes (in seconds) of the algorithms tested in solving the image deblurring problem.

F(x)	COMET, variant 1	COMET, variant 2	COMET, variant 3	AMGS	ACGM	FISTA CP	FISTA
45.74	1.33	1.21	1.87	2.52	1.76	1.92	2.16
25.61	2.77	2.35	3.14	3.98	3.45	3.57	3.67
13.22	4.19	3.78	4.52	6.21	4.93	5.23	5.84
5.83	5.49	4.98	6.02	9.42	6.76	7.38	7.69
3.25	6.97	5.89	7.32	13.21	8.35	9.21	9.84
1.11	8.29	7.82	8.75	17.65	10.79	12.41	12.73
0.63	9.72	9.46	10.06	22.08	13.24	15.86	16.25
0.51	11.14	11.31	12.69	26.39	15.65	17.13	17.97
0.44	13.53	13.93	14.21	34.11	17.23	19.32	20.15
0.37	15.86	16.56	16.72	41.28	19.86	23.57	24.43
0.35	17.30	18.27	18.96	49.36	25.57	28.39	32.07

were obtained by running the first variant of COMET in intervals of 20 iterations. The other entries in the table were obtained by computing the time spent by the other methods to achieve the same decrease in the values of the objective function. Analyzing the obtained results, we can observe that the different variants of the estimating sequences methods are very efficient. Different from the other estimating sequence methods, we can see that the performance of AMGS is significantly affected by the need to compute an additional proximal step per iteration. Comparing to FISTA, every variant of COMET and ACGM perform more computations per iteration. Nevertheless, we can see that the improvement in runtime is significant. Comparing among the estimating sequence methods, we can observe that the fastest variant of COMET converges approximately 30% faster than AMGS. Last, we note that the differences in runtime among all variants of COMET are marginal. Nevertheless, we note that the variant of COMET which is obtained by initializing $\gamma_0 = 0$ is more efficient, while also enjoying the robustness to the imperfect knowledge of the strong convexity parameter.

6. Conclusions and discussion

The problem of constructing accelerated black-box first-order methods for solving optimization problems with composite structure by utilizing the estimating sequences framework has been considered, and a new class of estimating functions has been introduced. It has been shown that by exploiting these estimating sequences together with the gradient mapping technique, it is possible to construct very efficient gradient-based methods, which we named COMET. Unlike the existing results on the convergence of FGM-type methods, the novel convergence analysis established in this work allows for the adaptation of the step-size. Another major contribution which stemmed from the proposed convergence analysis is the fact that COMET is guaranteed to converge when $\gamma_0 \in [0, 3L + \mu_{\hat{f}}]$. The practical implication of these two observations is the fact that it is possible to construct efficient accelerated methods which are also robust to the imperfect knowledge of the smoothness and strong convexity parameters. Our theoretical findings were corroborated by extensive numerical experiments, wherein both synthetic and real-world data were utilized.

The results that were established in this work can be further developed in different directions. Particularly, it is interesting to investigate the possibilities of embedding the heavy-ball momentum into COMET. Another attractive research direction is the investigation of the possibility of coupling between the proposed framework and the inexact oracle framework, as well as the framework for constructing distributed proximal gradient methods. Lastly, we note that it is also interesting to investigate the possible extensions to designing accelerated algorithms for solving non-convex optimization problems.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

CRediT authorship contribution statement

Endrit Dosti: Conceptualization, Formal analysis, Methodology, Software, Visualization, Writing – review & editing. **Sergiy A. Vorobyov:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – review & editing. **Themistoklis Charalambous:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Data Availability

Data will be made available on request.

Appendix A. Proof of Theorem 1

We start by showing that $m_L(y; x)$ is an L-strongly convex function in x. Notice that it is defined to be the sum of convex functions. Therefore, it is itself a convex function. Now, consider that

$$m_L(y; y) - m_L(y; T_L(y)) \geq \frac{L}{2}||y - T_L(y)||^2. \tag{41}$$

By the definition given in (9), $T_L(y)$ is the minimizer of $m_L(y; x)$ over all $x \in \mathcal{R}^n$. Therefore, we can conclude that $m_L(y; x)$ is a strongly convex function with strong convexity parameter L.

Now, we can proceed to deriving the lower bound. From (5), (6), it can be written that

$$F(x) \geq \hat{f}(y) + \tau\hat{g}(y) + \left(\nabla\hat{f}(y) + \tau s_L(y)\right)^T (x - y) + \frac{\mu_{\hat{f}}}{2}||x - y||^2. \tag{42}$$

Then, from the definition of $m_L(y, y)$ given in (7), as well as (12), the right-hand side (RHS) of (43) can be rewritten as

$$\begin{aligned} &\hat{f}(y) + \tau\hat{g}(y) + \left(\nabla\hat{f}(y) + \tau s_L(y)\right)^T (x - y) + \frac{\mu_{\hat{f}}}{2}\|x - y\|^2 \\ &= m_L(y; y) + r_L(y)^T (x - y) + \frac{\mu_{\hat{f}}}{2}\|x - y\|^2. \end{aligned} \tag{43}$$

Moreover, substituting (42) in (44), the lower bound of the RHS of (44) becomes

$$\begin{aligned} m_L(y; y) + r_L(y)^T (x - y) + \frac{\mu_{\hat{f}}}{2}\|x - y\|^2 \geq m_L(y; T_L(y)) \\ + \frac{L}{2}\|y - T_L(y)\|^2 + r_L(y)^T (x - y) + \frac{\mu_{\hat{f}}}{2}\|x - y\|^2. \end{aligned}$$

Utilizing the definition of the reduced composite gradient given in (10), yields

$$m_L(y;T_L(y))+\frac{L}{2}\|y-T_L(y)\|^2+r_L(y)^T(x-y)+\frac{\mu_{\hat{f}}}{2}\|x-y\|^2$$
$$=m_L(y;T_L(y))+\frac{1}{2L}\|r_L(y)\|^2+r_L(y)^T(x-y)+\frac{\mu_{\hat{f}}}{2}\|x-y\|^2. \quad (44)$$

Finally, taking a proximal gradient descent step on $f(x)$, which by assumption has Lipschitz continuous gradient, we can obtain (13). This completes the proof.

Appendix B. Proof of Lemma 1

By the assumption of Lemma 1, we have

$$F(x_k) \le \phi_k^* = \min_{x\in\mathcal{R}^n}\phi_k(x) \overset{(14)}{\le} \min_{x\in\mathcal{R}^n}[\lambda_k\phi_0(x)+(1-\lambda_k)F(x)]$$
$$\le \lambda_k\phi_0(x^*)+(1-\lambda_k)F(x^*).$$

Rearranging the terms yields the desired result.

Appendix C. Proof of Lemma 2

We prove this lemma by induction. Let us begin by analyzing iteration $k=0$. By assumption, we have $\lambda_0=1$. Utilizing (14), we obtain $\phi_0(x)\le\lambda_0\phi_0(x)+(1-\lambda_0)F(x)\equiv\phi_0(x)$. Then, assuming that (14) holds true at some iteration k, it can be written that

$$\phi_k(x)-(1-\lambda_k)F(x)\le\lambda_k\phi_0(x). \quad (45)$$

Substituting the bound obtained in Theorem 1, i.e., (13) in (16), we obtain

$$\phi_{k+1}(x)\le(1-\alpha_k)\phi_k(x)+\alpha_kF(x). \quad (46)$$

Then, adding and subtracting the same term to the RHS of (47), we reach

$$\phi_{k+1}(x) \le (1-\alpha_k)\phi_k(x)+\alpha_kF(x)+(1-\alpha_k)(1-\lambda_k)F(x)-(1-\alpha_k)$$
$$(1-\lambda_k)F(x)=(1-\alpha_k)[\phi_k(x)-(1-\lambda_k)F(x)]$$
$$+(\alpha_k+(1-\lambda_k)(1-\alpha_k))F(x). \quad (47)$$

Using the bound obtained in (46) in (48), we have

$$\phi_{k+1}(x)\le(1-\alpha_k)\lambda_k\phi_0(x)+(1-\lambda_k+\alpha_k\lambda_k)F(x). \quad (48)$$

Lastly, after utilizing (15), the proof is concluded.

Appendix D. Proof of Lemma 3

Let us begin with establishing the first part of the proof through a mathematical induction argument. At iteration $k=0$, we have $\nabla^2\phi_0(x)=\gamma_0 I$. Next, assuming that at some iteration k it is true that $\nabla^2\phi_k(x)=\gamma_k I$, at iteration $k+1$ it can be written that

$$\nabla^2\phi_{k+1}(x)\overset{(16)}{=}(1-\alpha_k)\gamma_kI+\alpha_k\mu_{\hat{f}}I\equiv\gamma_{k+1}I. \quad (49)$$

We then proceed to establishing the proposed recurrent relations for updating the terms in the sequences $\{v_k\}_{k=0}^{\infty}$ and $\{\phi_k^*\}_{k=0}^{\infty}$. Substituting (17) into (16), and analyzing its first-order optimality conditions we obtain

$$\gamma_{k+1}(x-v_{k+1})=\gamma_k(1-\alpha_k)(x-v_k)+\alpha_k\big(\mu_{\hat{f}}(x-y_k)r_{L_k}(y_k)\big). \quad (50)$$

We can then reduce the terms that depend on x by using (18) in (51), and reach

$$-\gamma_{k+1}v_{k+1} \quad =-(1-\alpha_k)\gamma_kv_k+\alpha_k\big(-\mu_{\hat{f}}y_k+r_{L_k}(y_k)\big). \quad (51)$$

Then, substituting (10) in (52), we obtain (19).

To establish (20), let us begin by substituting (17) in (16), now evaluated at the point $x=y_k$. This way we obtain

$$\phi_{k+1}^* \quad +\frac{\gamma_{k+1}}{2}\|y_k-v_{k+1}\|^2=(1-\alpha_k)\Big(\phi_k^*+\frac{\gamma_k}{2}\|y_k-v_k\|^2\Big)$$
$$+\alpha_k\Big(F\big(T_{L_k}(y_k)\big)+\frac{1}{2L_k}\|r_{L_k}(y_k)\|^2\Big). \quad (52)$$

We proceed by utilizing (19) to compute the second term in the left hand side (LHS) of (53). Consider the following

$$v_{k+1}-y_k=\frac{1}{\gamma_{k+1}}((1-\alpha_k)\gamma_kv_k+\alpha_k\mu_{\hat{f}}y_k$$
$$-\alpha_kL_k\big(y_k-T_{L_k}(y_k)\big)-\gamma_{k+1}y_k). \quad (53)$$

Then, utilizing (18) in (54), we obtain

$$v_{k+1}-y_k=\frac{1}{\gamma_{k+1}}((1-\alpha_k)\gamma_k(v_k-y_k)-\alpha_kL_k\big(y_k-T_{L_k}(y_k)\big). \quad (54)$$

Taking $||\cdot||^2$ of both sides in (55), yields

$$||y_k-v_{k+1}||^2=\frac{||(1-\alpha_k)\gamma_k(v_k-y_k)-\alpha_kL_k\big(y_k-T_{L_k}(y_k)\big)||^2}{\gamma_{k+1}^2}. \quad (55)$$

Finally, multiplying both sides of (56) by $\frac{\gamma_{k+1}}{2}$ and expanding the RHS, we reach

$$\frac{\gamma_{k+1}}{2}\|y_k-v_{k+1}\|^2=\frac{(1-\alpha_k)^2\gamma_k^2}{2\gamma_{k+1}}\|v_k-y_k\|^2+\frac{\alpha_k^2{L_k}^2}{2\gamma_{k+1}}\|y_k-T_{L_k}(y_k)\|^2$$
$$-\frac{2L_k\alpha_k(1-\alpha_k)\gamma_k}{2\gamma_{k+1}}(v_k-y_k)^T\nabla\big(y_k-T_{L_k}(y_k)\big). \quad (56)$$

Substituting (57) in (53), and making some straightforward algebraic manipulations, we obtain (20).

Appendix E. Proof of Theorem 2

Set $\phi_0^*=f(x_0)$. Then, considering (17) evaluated at iteration $k=0$ and $x=x_0$, we obtain $\phi_0(x_0)=f(x_0)+\frac{\gamma_0}{2}||x_0-v_0||^2$. In Algorithm 1, we initialize $v_0=x_0$, which is sufficient to guarantee that $f(x_0)\le\phi_0^*$ at step $k=0$. Moreover, recall that we designed the update rules of the proposed method to guarantee that $f(x_k)\le\phi_k^*$, $\forall k=1,2,\ldots$. Therefore, the necessary conditions for the results proved in Lemma 1 to be applied are satisfied.

Appendix F. Proof of Lemma 4

Let $\gamma_0\in[0,3L_0+\mu_{\hat{f}}]$ and consider applying (18) to the following

$$\gamma_{k+1}-\mu_{\hat{f}}=(1-\alpha_k)\gamma_k+\alpha_k\mu_{\hat{f}}-\mu_{\hat{f}}. \quad (57)$$

Then, utilizing the assumption that $\lambda_0=1$ in (58), it can be written that

$$\gamma_{k+1}-\mu_{\hat{f}}=(1-\alpha_k)\lambda_0\big[\gamma_k-\mu_{\hat{f}}\big]. \quad (58)$$

Using the recursivity of (18) in (59), yields

$$\gamma_{k+1}-\mu_{\hat{f}}=\lambda_{k+1}\big[\gamma_0-\mu_{\hat{f}}\big]. \quad (59)$$

Let us now exploit the connection between relations (15) and (25), which can be linked through the term α_k as follows

$$\alpha_k=1-\frac{\lambda_{k+1}}{\lambda_k}=\sqrt{\frac{\gamma_{k+1}}{L_k}}=\sqrt{\frac{\mu_{\hat{f}}}{L_k}+\frac{\gamma_{k+1}-\mu_{\hat{f}}}{L_k}}. \quad (60)$$

Substituting (60) in the RHS of (61) and making some manipulations, we get

$$\frac{1}{\lambda_{k+1}}-\frac{1}{\lambda_k}=\frac{1}{\sqrt{\lambda_{k+1}}}\sqrt{\frac{\mu_{\hat{f}}}{\lambda_{k+1}L_k}+\frac{\gamma_0-\mu_{\hat{f}}}{L_k}}. \quad (61)$$

Then, through a difference of squares argument, we reach

$$\left(\frac{1}{\sqrt{\lambda_{k+1}}}-\frac{1}{\sqrt{\lambda_k}}\right)\left(\frac{1}{\sqrt{\lambda_{k+1}}}+\frac{1}{\sqrt{\lambda_k}}\right) = \frac{1}{\sqrt{\lambda_{k+1}}}\sqrt{\frac{\mu_{\hat{f}}}{\lambda_{k+1}L_k}+\frac{\gamma_0-\mu_{\hat{f}}}{L_k}}. \tag{62}$$

Let us now analyze the behavior of the terms in the sequence $\{\lambda_k\}_{k=0}^{\infty}$. First, recall that from Lemma 2 we have $\alpha_k \in [0,1]$. Then, considering (15), we can conclude that the terms λ_k are non-increasing in the iteration counter k. Therefore, we can substitute the term $\frac{1}{\sqrt{\lambda_k}}$ in the LHS of (63) with the larger number $\frac{1}{\sqrt{\lambda_{k+1}}}$. This results in

$$\frac{2}{\sqrt{\lambda_{k+1}}}\left(\frac{1}{\sqrt{\lambda_{k+1}}}-\frac{1}{\sqrt{\lambda_k}}\right) \geq \frac{1}{\sqrt{\lambda_{k+1}}}\sqrt{\frac{\mu_{\hat{f}}}{\lambda_{k+1}L_k}+\frac{\gamma_0-\mu_{\hat{f}}}{L_k}} \tag{63}$$

Note that the practical performance of the proposed method depends on the initialization of the parameter γ_0. To allow for the widest possible range of selection for this parameter, we need to consider separately the regions $\mathcal{R}_1=[0,\mu_{\hat{f}}[$ and $\mathcal{R}_2=[\mu_{\hat{f}},3L_k+\mu_{\hat{f}}]$. The results for the case when $\gamma_0\in\mathcal{R}_2$ can be established by following the analysis conducted for FGM in Nesterov [51, Lemma 2.2.4]. Therefore, in the sequel we will thoroughly establish the results only for the case when $\gamma_0\in\mathcal{R}_1$, which is the novel part of the proof. Let us begin by defining the following quantity

$$\xi_{k,\mathcal{R}_1} \triangleq \sqrt{\frac{L_{\max}}{(\mu_{\hat{f}}-\gamma_0)\lambda_k}}, \tag{64}$$

where $L_{\max}$ was defined in (32). Next, (64) can be rewritten as

$$\frac{2}{\sqrt{\lambda_{k+1}}}-\frac{2}{\sqrt{\lambda_k}} \geq \sqrt{\frac{\mu_{\hat{f}}-\gamma_0}{L_k}}\sqrt{\frac{\mu_{\hat{f}}L_k}{L_k\lambda_{k+1}(\mu_{\hat{f}}-\gamma_0)}-1}. \tag{65}$$

Then, relaxing the bound in (66) and multiplying it with $\sqrt{\frac{L_{\max}}{\mu_{\hat{f}}-\gamma_0}}$, we obtain

$$\xi_{k+1,\mathcal{R}_1}-\xi_{k,\mathcal{R}_1} \geq \frac{1}{2}\sqrt{\frac{\mu_{\hat{f}}\xi_{k+1,\mathcal{R}_1}^2}{L_{\max}}-1}. \tag{66}$$

We then proceed to establish via induction the following lower bound

$$\xi_{k,\mathcal{R}_1} \geq \frac{\sqrt{2}}{4\delta}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left[e^{(k+1)\delta}-e^{(k+1)\delta}\right], \tag{67}$$

where $\delta \triangleq \frac{1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_{\max}}}$. Utilizing (65) at step $k=0$, we have

$$\xi_{0,\mathcal{R}_1} = \sqrt{\frac{L_{\max}}{(\mu_{\hat{f}}-\gamma_0)\lambda_0}} = \sqrt{\frac{L_{\max}}{\mu_{\hat{f}}-\gamma_0}}, \tag{68}$$

where the second equality is obtained because $\lambda_0=1$. Then, substituting (32) into (69), we obtain

$$\xi_{0,\mathcal{R}_1} \geq \frac{\sqrt{2}}{2}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left[e^{1/2}-e^{-1/2}\right] \geq \frac{\sqrt{2}}{4\delta}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left[e^{\delta}-e^{-\delta}\right]. \tag{69}$$

Note that the second row in (70) follows because the RHS is increasing in δ, which by construction is always $\delta<0.5$.

As it is common with induction-type of proofs, the next step is to assume that (68) is satisfied for some iteration k. To establish that the relation would hold true at the next iteration as well, we proceed via contradiction. Define $\omega(t)\triangleq\frac{\sqrt{2}}{4\delta}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left[e^{(t+1)\delta}-e^{-(t+1)\delta}\right]$, and note that from Nesterov [51, Lemma 2.2.4] it is a convex function. Therefore, it can be written that

$$\omega(t) \leq \xi_{k,\mathcal{R}_1} \overset{(67)}{\leq} \xi_{k+1,\mathcal{R}_1}-\frac{1}{2}\sqrt{\frac{\mu_{\hat{f}}\xi_{k+1,\mathcal{R}_1}^2}{L_{\max}}-1}. \tag{70}$$

Assuming that $\xi_{k+1,\mathcal{R}_1}<\omega(t+1)$ and substituting it into (71), we have

$$\omega(t) < \omega(t+1)-\frac{1}{2}\sqrt{\frac{\mu_{\hat{f}}\xi_{k+1,\mathcal{R}_1}^2}{L_{\max}}-1}. \tag{71}$$

Then, utilizing the definition of δ, as well as (68), we obtain

$$\begin{aligned}\omega(t) &\leq \omega(t+1)-\frac{1}{2}\sqrt{4\delta^2\left[\frac{\sqrt{2}}{4\delta}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left(e^{(t+2)\delta}-e^{-(t+2)\delta}\right)\right]^2-1}\\ &\leq \omega(t+1)-\frac{\sqrt{2}}{4}\sqrt{\frac{L_k}{\mu_{\hat{f}}-\gamma_0}}\left[e^{(t+2)\delta}+e^{-(t+2)\delta}\right]=\omega(t+1)\\ &\quad+\omega'(t+1)(t-(t+1))\leq\omega(t),\end{aligned} \tag{72}$$

where the last inequality follows from the supporting hyperplane theorem of convex functions. Notice that this result contradicts the earlier assumption that $\xi_{k+1,\mathcal{R}_1}<\omega(t+1)$. Thus, the inductive argument asserts that we have established the lower bound (68) to be true for all values of $k=0,1,\ldots$.

We are finally ready to establish (34). From (65), it can be written that

$$\lambda_k = \frac{L_{\max}}{\xi_{k+1,\mathcal{R}_1}^2(\mu_{\hat{f}}-\gamma_0)}. \tag{73}$$

Utilizing (68) in the RHS of (74), we reach

$$\lambda_k \leq \frac{(4\delta)^2L_{\max}}{2L_k\left[e^{(k+1)\delta}-e^{(k+1)\delta}\right]^2}, \tag{74}$$

The first inequality in (34) is obtained by substituting the definition of δ in (75).

To establish the remaining inequality in (34), we first analyze the following

$$\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2 = e^{(k+1)\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-e^{-(k+1)\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-2. \tag{75}$$

Then, utilizing the definition of the hyperbolic cosine function in (76), we obtain

$$\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2 = 2\cosh\left(\sqrt{\frac{\mu_{\hat{f}}}{L_k}}(k+1)-2\right). \tag{76}$$

Using the Taylor expansion of the hyperbolic cosine function, yields

$$\begin{aligned}\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2 &= -2+2+2\frac{\mu_{\hat{f}}(k+1)^2}{2L_k}\\ &\quad+2\frac{\mu_{\hat{f}}^2(k+1)^4}{4!{L_k}^2}+\ldots.\end{aligned} \tag{77}$$

The next step is to truncate the RHS of (78). This results in

$$\left(e^{\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}-e^{-\frac{k+1}{2}\sqrt{\frac{\mu_{\hat{f}}}{L_k}}}\right)^2 \geq \frac{\mu_{\hat{f}}}{L_k}(k+1)^2. \tag{78}$$

All that remains for establishing the second inequality of (34), is to substitute (79) into the denominator of the first inequality of (34).

Appendix G. Proof of Lemma 5

We begin by substituting the upper bound (4) evaluated at the point $y = x^*$ into (3), and obtain that

$$F(x_0) = \hat{f}(x_0) + \tau\hat{g}(x_0) \leq \hat{f}(x^*) + \nabla\hat{f}(x^*)^T(x_0 - x^*) + \frac{L_0}{2}||x_0 - x^*||^2 + \tau\hat{g}(x_0). \tag{79}$$

Then, from the equality established in (12), the RHS of (80) can be written as

$$\begin{aligned} F(x_0) \leq\ & \hat{f}(x^*) + \nabla\hat{f}(x^*)^T(x_0 - x^*) + \frac{L_0}{2}\|x_0 - x^*\|^2 + \tau\hat{g}(x_0) = \hat{f}(x^*) \\ & + \Big(\tau s_{L_0}(x^*) - L_0\big(x^* - T_{L_0}(x^*)\big)\Big)^T(x_0 - x^*) + \frac{L_0}{2}\|x_0 - x^*\|^2 \\ & + \tau\hat{g}(x_0). \end{aligned} \tag{80}$$

From the definition of the composite gradient mapping given in (9), we can see that when $y = x^*$, then $T_{L_0}(x^*) = x^*$. Therefore, the RHS of (81) becomes

$$F(x_0) \leq \hat{f}(x^*) - \tau s_{L_0}(x^*)^T(x^* - x_0) + \frac{L_0}{2}||x_0 - x^*||^2 + \tau\hat{g}(x_0). \tag{81}$$

Lastly, utilizing (6) in the RHS of (82) completes the proof.

References

[1] Y. Nesterov, Gradient methods for minimizing composite objective function, Math. Program. 140 (1) (Aug. 2013) 125–161.
[2] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (Jan. 2014) 127–239.
[3] V. Cevher, S. Becker, M. Schmidt, Convex optimization for big data: scalable, randomized, and parallel algorithms for big data analytics, IEEE Signal Process. Mag. 31 (5) (Aug. 2014) 32–43.
[4] K. Slavakis, G.B. Giannakis, G. Mateos, Modeling and optimization for big data analytics: (statistical) learning tools for our era of data deluge, IEEE Signal Process. Mag. 31 (5) (Aug. 2014) 18–31.
[5] A.P. Liavas, G. Kostoulas, G. Lourakis, K. Huang, N.D. Sidiropoulos, Nesterov-based alternating optimization for nonnegative tensor factorization: algorithm and parallel implementation, IEEE Trans. Signal Process. 66 (4) (Nov. 2018) 944–953.
[6] M.S. Ibrahim, A. Konar, N.D. Sidiropoulos, Fast algorithms for joint multicast beamforming and antenna selection in massive MIMO, IEEE Trans. Signal Process. 68 (Mar. 2020) 1897–1909.
[7] R. Gu, A. Dogandžić, Projected Nesterov's proximal-gradient algorithm for sparse signal recovery, IEEE Trans. Signal Process. 65 (13) (May. 2017) 3510–3525.
[8] K. Elkhalil, A. Kammoun, X. Zhang, M. Alouini, T. Al-Naffouri, Risk convergence of centered kernel ridge regression with large dimensional data, IEEE Trans. Signal Process. 68 (Feb. 2020) 1574–1588.
[9] M.J. Wainwright, Structured regularizers for high-dimensional problems: statistical and computational issues, Annu. Rev. Stat. Appl. 1 (1) (Jan. 2014) 233–253.
[10] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. 58 (1) (Jan. 1996) 267–288.
[11] J. Tropp, S.J. Wright, Computational methods for sparse solution of linear inverse problems, Proc. IEEE 98 (6) (Apr. 2010) 948–958.
[12] P.L. Combettes, J.C. Pesquet, Proximal splitting methods in signal processing, Fixed-Point Algorithms Inverse Probl.Sci. Eng. 49 (May. 2011) 185–212.
[13] E.J. Candàs, Y.C. Eldar, T. Strohmer, V. Voroninski, Phase retrieval via matrix completion, SIAM Rev. 57 (2) (May. 2015) 225–251.
[14] A. Yurtsever, Y.P. Hsieh, V. Cevher, Scalable convex methods for phase retrieval, in: Proc. IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico, Dec. 2015, pp. 381–384.
[15] C. Studer, T. Goldstein, W. Yin, R.G. Baraniuk, Democratic representations, arXiv:1401.3420(Apr. 2015).
[16] Y. Nesterov, Subgradient methods for huge-scale optimization problems, Math. Program. 146 (1) (Aug. 2014) 275–297.
[17] A. Beck, First-order Methods in Optimization, vol. 25, SIAM, Oct. 2017.
[18] A. d'Aspremont, D. Scieur, A. Taylor, Acceleration methods, Found. Trends Optim. 5 (1–2) (Dec. 2021) 1–245.
[19] A. Nemirovsky, D. Yudin, Problem Complexity and Method Efficiency in Optimization, Wiley, 1983.
[20] B.T. Polyak, Some methods of speeding up the convergence of iteration methods, USSR Comput. Math. Math. Phys. 4 (5) (1964) 1–17.
[21] Y. Nesterov, A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$, Doklady USSR 269 (1983) 543–547.
[22] A. Auslender, M. Teboulle, Interior gradient and proximal methods for convex and conic optimization, SIAM J. Optim. 16 (3) (Jul. 2006) 697–725.
[23] G. Lan, Z. Lu, R.D.C. Monteiro, Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming, Math. Program. 126 (1) (Jan. 2011) 1–29.
[24] B. O'donoghue, E. Candès, Adaptive restart for accelerated gradient schemes, Found. Comput. Math. 15 (3) (Jun. 2015) 715–732.
[25] A. d'Aspremont, Smooth optimization with approximate gradient, SIAM J. Optim. 19 (3) (Oct. 2008) 1171–1183.
[26] M. Schmidt, N.L. Roux, F.R. Bach, Convergence rates of inexact proximal-gradient methods for convex optimization, in: Proc. 25th Annual Conference on Neural Information Processing Systems, Granada, Spain, 2011, pp. 1458–1466.
[27] O. Devolder, F. Glineur, Y. Nesterov, First-order methods of smooth convex optimization with inexact oracle, Math. Program. 146 (1) (Aug. 2014) 37–75.
[28] Z. Allen-Zhu, L. Orecchia, Linear coupling: an ultimate unification of gradient and mirror descent, Nov. 2016, arXiv:1407.1537.
[29] S. Bubeck, Y.T. Lee, M. Singh, A geometric alternative to Nesterov's accelerated gradient descent, Jun. 2015. arXiv:1506.08187.
[30] N. Flammarion, F. Bach, From averaging to acceleration, there is only a step-size, in: Proc. Conference on Learning Theory, Paris, France, 2015, pp. 658–695.
[31] W. Su, S. Boyd, E.J. Candès, A differential equation for modeling Nesterov's accelerated gradient method: theory and insights, J. Mach. Learn. Res. 17 (153) (Jan. 2016) 1–43.
[32] L. Lessard, B. Recht, A. Packard, Analysis and design of optimization algorithms via integral quadratic constraints, SIAM J. Optim. 26 (1) (Jan. 2016) 57–95.
[33] Y. Drori, M. Teboulle, Performance of first-order methods for smooth convex minimization: a novel approach, Math. Program. 145 (1) (Jun. 2014) 451–482.
[34] D. Kim, J.A. Fessler, Optimized first-order methods for smooth convex minimization, Math. Program. 159 (1) (Sep. 2016) 81–107.
[35] D. Kim, J.A. Fessler, Generalizing the optimized gradient method for smooth convex minimization, SIAM J. Optim. 28 (2) (Jun. 2018) 1920–1950.
[36] S. Bubeck, Convex optimization: algorithms and complexity, Found. Trends Mach. Learn. (May. 2014) 231–357.
[37] M.I. Florea, S.A. Vorobyov, An accelerated composite gradient method for large-scale composite objective problems, IEEE Trans. Signal Process. 67 (2) (Jan. 2019) 444–459.
[38] Y. Nesterov, Universal gradient methods for convex optimization problems, Math. Program. 152 (1) (Aug. 2015) 381–404.
[39] Y. Nesterov, Accelerating the cubic regularization of Newton's method on convex problems, Math. Program. 112 (1) (Mar. 2008) 159–181.
[40] X. Chen, B. Jiang, T. Lin, S. Zhang, Accelerating adaptive cubic regularization of Newton's method via random sampling, J. Mach. Learn. Res. 23 (Mar. 2022) 1–38.
[41] Y. Nesterov, Inexact high-order proximal-point methods with auxiliary search procedure, SIAM J. Optim. 31 (4) (Nov. 2021) 2807–2828.
[42] N. Doikov, Y. Nesterov, High-order optimization methods for fully composite problems, SIAM J. Optim. 32 (3) (Sep. 2022) 2402–2427.
[43] X. Zeng, J. Lei, J. Chen, Dynamical primal-dual accelerated method with applications to network optimization, IEEE Trans. Autom. Control (2022), doi:10.1109/TAC.2022.3152720. (Early Access)
[44] J. Gao, X. Liu, Y. Dai, Y. Huang, P. Yang, A family of distributed momentum methods over directed graphs with linear convergence, IEEE Trans. Autom. Control (2022), doi:10.1109/TAC.2022.3160684. (Early Access)
[45] S.S. Mannelli, P. Urbani, Analytical study of momentum-based acceleration methods in paradigmatic high-dimensional non-convex problems, Adv. Neural Inf. Process. Syst. 34 (Dec. 2021) 187–199.
[46] X. Xie, P. Zhou, H. Li, Z. Lin, S. Yan, ADAN: adaptive Nesterov momentum algorithm for faster optimizing deep models, Aug. 2022, arXiv:2208.06677.
[47] A. Kulunchakov, J. Mairal, Estimate sequences for stochastic composite optimization: variance reduction, acceleration, and robustness to noise, J. Mach. Learn. Res. 21 (155) (Jul. 2020) 1–52.
[48] M. Even, R. Berthier, F. Bach, N. Flammarion, P. Gaillard, H. Hendrikx, L. Massoulié, A. Taylor, A continuized view on Nesterov acceleration for stochastic gradient descent and randomized gossip, Jun. 2021. arXiv:2106.07644.
[49] K. Ahn, S. Sra, From Nesterov's estimate sequence to Riemannian acceleration, in: Proc. Conference on Learning Theory, Graz, Austria, 2020, pp. 88–118.
[50] J. Kim, I. Yang, Nesterov acceleration for Riemannian optimization, Feb. 2022, arXiv:2202.02036.
[51] Y. Nesterov, Lectures on Convex Optimization, vol. 137, Springer, Dec. 2018.
[52] M. Baes, Estimate sequence methods: extensions and approximations, Institute for Operations Research, ETH, Zürich, Switzerland, Aug. 2009.
[53] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (Mar. 2009) 183–202.
[54] M.I. Florea, S.A. Vorobyov, A generalized accelerated composite gradient method: uniting Nesterov's fast gradient method and FISTA, IEEE Trans. Signal Process. 68 (Jul. 2020) 3033–3048.
[55] E. Dosti, S.A. Vorobyov, T. Charalambous, Embedding a heavy-ball type of momentum into the estimating sequences, Aug. 2020, arXiv:2008.07979.
[56] E. Dosti, S.A. Vorobyov, T. Charalambous, Generalizing Nesterov's acceleration framework by embedding momentum into estimating sequences: new algorithm and bounds, in: IEEE International Symposium on Information Theory (ISIT), Helsinki, Finland, Jun 2022, pp. 1506–1511.

[57] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, online available at https://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf.
[58] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (May. 2011) 1–27.
[59] M. Grant, S. Boyd, Y. Ye, CVX: Matlab software for disciplined convex programming (web page and software), 2009.
[60] S.R. Becker, E.J. Candès, M. Grant, Templates for convex cone problems with applications to sparse signal recovery, Math. Program. 3 (3) (2011) Sep.165.
[61] A. Chambolle, T. Pock, An introduction to continuous optimization for imaging, Acta Numer. 25 (2016) 161–319.