HHS Public Access

Author manuscript

SIAM J Optim. Author manuscript; available in PMC 2018 May 23.

Published in final edited form as:

SIAM J Optim. 2018 ; 28(1): 223–250. doi:10.1137/16M108940X.

ANOTHER LOOK AT THE FAST ITERATIVE SHRINKAGE/ THRESHOLDING ALGORITHM (FISTA)*

Donghwan Kim† and **Jeffrey A. Fessler**†

†Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA

Abstract

This paper provides a new way of developing the "Fast Iterative Shrinkage/Thresholding Algorithm (FISTA)" [3] that is widely used for minimizing composite convex functions with a nonsmooth term such as the ℓ_1 regularizer. In particular, this paper shows that FISTA corresponds to an optimized approach to accelerating the proximal gradient method with respect to a worst-case bound of the cost function. This paper then proposes a new algorithm that is derived by instead optimizing the step coefficients of the proximal gradient method with respect to a worst-case bound of the composite gradient mapping. The proof is based on the worst-case analysis called Performance Estimation Problem in [11].

1. Introduction

The "Fast Iterative Shrinkage/Thresholding Algorithm" (FISTA) [3], also known as a fast proximal gradient method (FPGM) in general, is a very widely used fast first-order method. FISTA's speed arises from Nesterov's accelerating technique in [23, 24] that improves the $O(1/N)$ cost function worst-case bound of a proximal gradient method (PGM) to the optimal $O(1/N^2)$ rate where N denotes the number of iterations [3].

This paper first provides a new way to develop Nesterov's acceleration approach, *i.e.*, FISTA (FPGM). In particular, we show that FPGM corresponds to an optimized approach to accelerating PGM with respect to a worst-case bound of the cost function. We then propose a new fast algorithm that is derived from PGM by instead optimizing a worst-case bound of the composite gradient mapping. We call this new method FPGM-OCG (OCG for optimized over composite gradient mapping). This new method provides the best known analytical worst-case bound for decreasing the composite gradient mapping with rate $O(1/N^{\frac{3}{2}})$ among fixed-step first-order methods. The proof is based on the worst-case bound analysis called Performance Estimation Problem (PEP) in [11], which we briefly review next.

Drori and Teboulle's PEP [11] casts a worst-case analysis for a given optimization method and a given class of optimization problems into a meta-optimization problem. The original PEP has been intractable to solve exactly, so [11] introduced a series of tractable relaxations,

*Submitted to the editors July 31, 2017.

focusing on first-order methods and smooth convex minimization problems; this PEP and its relaxations were studied for various algorithms and minimization problem classes in [12, 16, 17, 18, 19, 29, 30]. Drori and Teboulle [11] further proposed to optimize the step coefficients of a given class of optimization methods using a PEP. This approach was studied for first-order methods on unconstrained smooth convex minimization problems in [11], and the authors [17] derived a new first-order method, called an optimized gradient method (OGM) that has an analytic worst-case bound on the cost function that is twice smaller than the previously best known bounds of [23, 24]. Recently, Drori [10] showed that the OGM exactly achieves the optimal cost function worst-case bound among first-order methods for smooth convex minimization (in high-dimensional problems).

Building upon [11] and its successors, Taylor *et al.* [29] expanded the use of PEP to first-order (proximal gradient) methods for minimizing nonsmooth composite convex functions. They used a tight relaxation[1] for PEP and studied the tight (exact) numerical worst-case bounds of FPGM, a proximal gradient version of OGM, and some variants versus number of iterations N. Their numerical results suggest that there exists an OGM-type acceleration of PGM that has a worst-case cost function bound that is about twice smaller than that of FPGM, showing room for improvement in accelerating PGM. However, it is difficult to derive an analytical worst-case bound for the tightly relaxed PEP in [29], so optimizing the step coefficients of PGM remains an open problem, unlike [11, 17] for smooth convex minimization.

Different from the tightly relaxed PEP in [29], this paper suggests a new (looser) relaxation of a cost function form of PEP for nonsmooth composite convex minimization that simplifies analysis and optimization of step coefficients of PGM, although yields loose worst-case bounds. Interestingly, the resulting optimized PGM numerically appears to be the FPGM. Then, we further provide a new generalized version of FPGM using our relaxed PEP that extends our understanding of the FPGM variants.

This paper next extends the PEP analysis of the gradient norm in [29, 30]. For unconstrained smooth convex minimization, the authors [16] used such PEP to optimize the step coefficients with respect to the gradient norm. The corresponding optimized algorithm can be useful particularly when dealing with dual problems where the gradient norm decrease is important in addition to the cost function minimization (see *e.g.*, [9, 22, 26]). By extending [16], this paper optimizes the step coefficients of the PGM for the composite gradient mapping form of PEP for nonsmooth composite convex minimization. The resulting optimized algorithm differs somewhat from Nesterov's acceleration and turns out to belong to the proposed generalized FPGM class.

Sec. 2 describes a nonsmooth composite convex minimization problem and first-order (proximal gradient) methods. Sec. 3 proposes a new relaxation of PEP for nonsmooth composite convex minimization problems and the proximal gradient methods, and suggests that the FPGM (FISTA) [3] is the optimized method of the cost function form of this relaxed

[1]Tight relaxation here denotes transforming (relaxing) an optimization problem into a solvable problem while their solutions remain the same. [29] tightly relaxes the PEP into a solvable equivalent problem under a large-dimensional condition.

PEP. Sec. 3 further proposes a generalized version of FPGM using the relaxed PEP. Sec. 4 studies the composite gradient mapping form of the relaxed PEP and describes a new optimized method for decreasing the norm of composite gradient mapping. Sec. 5 compares the various algorithms considered, and Sec. 6 concludes.

2. Problem, methods, and contribution

We consider first-order algorithms for solving the nonsmooth composite convex minimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \{F(\boldsymbol{x}) := f(\boldsymbol{x}) + \phi(\boldsymbol{x})\}, \quad \text{(M)}$$

under the following assumptions:

- $f: \mathbb{R}^d \to \mathbb{R}$ is is a convex function of the type $\mathcal{C}_L^{1,1}(\mathbb{R}^d)$, i.e., continuously differentiable with Lipschitz continuous gradient:

$$\| \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) \| \le L \| \boldsymbol{x} - \boldsymbol{y} \|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \quad (1)$$

  where $L > 0$ is the Lipschitz constant.
- $\phi: \mathbb{R}^d \to \mathbb{R}$ is proper, closed, convex and "proximal-friendly" [6].
- The optimal set $X_*(F) = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x})$ is nonempty, *i.e.*, the problem (M) is solvable.

We use $\mathcal{F}_L(\mathbb{R}^d)$ to denote the class of functions F that satisfy the above conditions. We additionally assume that the distance between the initial point $\boldsymbol{x}_0$ and an optimal solution $\boldsymbol{x}_* \in X(F)$ is bounded by $R > 0$, i.e., $\|\boldsymbol{x}_0 - \boldsymbol{x}_*\| \le R$.

PGM is a standard first-order method for solving the problem (M) [3, 6], particularly when the following proximal gradient update (that consists of a gradient descent step and a proximal operation [6]) is relatively simple:

$$\mathbf{p}_L(\boldsymbol{y}) := \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y}) \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \phi(\boldsymbol{x}) \right\} = \arg\min_{\boldsymbol{x}} \left\{ \frac{L}{2} \left\| \boldsymbol{x} - \left(\boldsymbol{y} - \frac{1}{L} \nabla f(\boldsymbol{y}) \right) \right\|^2 + \phi(\boldsymbol{x}) \right\}. \quad (2)$$

For $\phi(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, the update (2) becomes a simple shrinkage/thresholding update, and PGM reduces to an iterative shrinkage/thresholding algorithm (ISTA) [8]. (See [6, Table 10.2] for more functions $\phi(\boldsymbol{x})$ that lead to simple proximal operations.) PGM has the following bound on the cost function [3, Thm. 3.1] for any $N \ge 1$:

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \leq \frac{LR^2}{2N}. \quad (3)$$

Algorithm PGM

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{x}_i)$$

For simplicity in later derivations, we use the following definition of the composite gradient mapping [27]:

$$\tilde{\nabla}_L F(\boldsymbol{x}) := -L(\mathbf{p}_L(\boldsymbol{x}) - \boldsymbol{x}). \quad (4)$$

The composite gradient mapping reduces to the usual function gradient $\nabla f(\boldsymbol{x})$ when $\phi(\boldsymbol{x}) = 0$. We can then rewrite the PGM update in the following form reminiscent of a gradient method:

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{x}_i) = \boldsymbol{x}_i - \frac{1}{L}\tilde{\nabla}_L F(\boldsymbol{x}_i), \quad (5)$$

where each update guarantees the following monotonic cost function descent [27, Thm. 1]:

$$F(\boldsymbol{x}_i) - F(\boldsymbol{x}_{i+1}) \geq \frac{1}{2L}\|\tilde{\nabla}_L F(\boldsymbol{x}_i)\|^2. \quad (6)$$

For any $\boldsymbol{x} \in \mathbb{R}^d$, there exists a subgradient $\phi'(\mathbf{p}_L(\boldsymbol{x})) \in \partial\phi(\mathbf{p}_L(\boldsymbol{x}))$ that satisfies the following equality [3, Lemma 2.2]:

$$\tilde{\nabla}_L F(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) + \phi'(\mathbf{p}_L(\boldsymbol{x})). \quad (7)$$

This equality implies that any point $\bar{\boldsymbol{x}}$ with a zero composite gradient mapping ($\tilde{\nabla}_L F(\bar{\boldsymbol{x}}) = 0$, *i.e.*, $\bar{\boldsymbol{x}} = \mathbf{p}_L(\bar{\boldsymbol{x}})$) satisfies $0 \in \partial F(\bar{\boldsymbol{x}})$ and is a minimizer of (M). As discussed, minimizing the composite gradient mapping is noteworthy in addition to decreasing the cost function. This property becomes particularly important when dealing with dual problems. In particular, it is known that the norm of the dual (sub)gradient is related to the primal feasibility (see *e.g.*, [9, 22, 26]). Furthermore, the norm of the subgradient is upper bounded by the norm of the

composite gradient mapping, *i.e.*, for any given subgradients $\phi'(\mathbf{p}_L(\boldsymbol{x}))$ in (7) and $F'(\mathbf{p}_L(\boldsymbol{x})) := \nabla f(\mathbf{p}_L(\boldsymbol{x})) + \phi'(\mathbf{p}_L(\boldsymbol{x})) \in \partial F(\mathbf{p}_L(\boldsymbol{x}))$, we have

$$\|F'(\mathbf{p}_L(\boldsymbol{x}))\| \le \|\nabla f(\boldsymbol{x}) - \nabla f(\mathbf{p}_L(\boldsymbol{x}))\| + \|\nabla f(\boldsymbol{x}) + \phi'(\mathbf{p}_L(\boldsymbol{x}))\| \le 2L\|\boldsymbol{x} - \mathbf{p}_L(\boldsymbol{x})\| = 2\|\widetilde{\nabla}_L F(\mathbf{p}_L(\boldsymbol{x}))\|, \tag{8}$$

where the first inequality uses the triangle inequality and the second inequality uses (1) and (7). This inequality provides a close relationship between the primal feasibility and the dual composite gradient mapping. Therefore, we next analyze the worst-case bound of the composite gradient mapping of PGM; Sec. 4 below discusses a first-order algorithm that is optimized with respect to the composite gradient mapping.[2]

The following lemma shows that PGM monotonically decreases the norm of the composite gradient mapping.

Lemma 1

The PGM monotonically decreases the norm of composite gradient mapping, *i.e.*, for all $\boldsymbol{x}$:

$$\|\widetilde{\nabla}_L F(\mathbf{p}_L(\boldsymbol{x}))\| \le \|\widetilde{\nabla}_L F(\boldsymbol{x})\|. \tag{9}$$

Proof—The proof in [22, Lemma 2.4] can be easily extended to prove (9) using the nonexpansiveness of the proximal mapping (proximity operator) [6].

The following theorem provides a $O(1/N)$ bound on the norm of composite gradient mapping for the PGM, using the idea in [26] and Lemma 1.

Theorem 2

Let $F: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by PGM. Then for $N \ge 2$,

$$\min_{i \in \{0,\ldots,N\}} \|\widetilde{\nabla}_L F(\boldsymbol{x}_i)\| = \|\widetilde{\nabla}_L F(\boldsymbol{x}_N)\| \le \frac{2LR}{\sqrt{(N-1)(N+2)}}. \tag{10}$$

Proof—Let $m = \lfloor \frac{N}{2} \rfloor$, and we have

[2]One could develop a first-order algorithm that is optimized with respect to the norm of the subgradient (rather than its upper bound in Sec. 4), which we leave as future work.

$$\frac{LR^2}{2m} \overset{(3)}{\geq} F(\boldsymbol{x}_m) - F(\boldsymbol{x}_*) \overset{(6)}{\geq} F(\boldsymbol{x}_{N+1}) - F(\boldsymbol{x}_*) + \frac{1}{2L}\sum_{i=m}^{N} \|\widetilde{\nabla}_L F(\boldsymbol{x}_i)\|^2 \overset{(9)}{\geq} \frac{N-m+1}{2L}\|\widetilde{\nabla}_L F(\boldsymbol{x}_N)\|^2,$$

which is equivalent to (10) using $m \geq \frac{N-1}{2}$ and $N - m \geq \frac{N}{2}$.

Despite its inexpensive per-iteration computational cost, PGM suffers from the slow rate $O(1/N)$ for decreasing both the cost function and the norm of composite gradient mapping.[3] Therefore for acceleration, this paper considers the following class of *fixed-step* first-order methods (FSFOM), where the $(i+1)$th iteration consists of one proximal gradient evaluation, just like PGM, and a weighted summation of previous and current proximal gradient updates $\{\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\}_{k=0}^{i}$ with step coefficients $\{h_{i+1,k}\}_{k=0}^{i}$.

Algorithm Class FSFOM

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i) = \boldsymbol{y}_i - \frac{1}{L}\widetilde{\nabla}_L F(\boldsymbol{y}_i)$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \sum_{k=0}^{i} h_{i+1,k}(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k) = \boldsymbol{y}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k}\widetilde{\nabla}_L F(\boldsymbol{y}_k).$$

Although the weighted summation in FSFOM seems at first to be inefficient both computationally and memory-wise, the optimized FSFOM presented in this paper have equivalent recursive forms that have memory and computation requirements that are similar to PGM. Note that this class FSFOM includes PGM but excludes accelerated algorithms in [13, 25, 27] that combine the proximal operations and the gradient steps in other ways.

Among FSFOM[4], FISTA [3], also known as FPGM, is widely used since it has computation and memory requirements that are similar to PGM yet it achieves the optimal $O(1/N^2)$

[3] [11, Thm. 2] and [16, Thm. 2] imply that the $O(1/N)$ rates of both the cost function bound (3) and the composite gradient mapping norm bound (10) of PGM are tight up to a constant respectively.

[4] The step coefficients of FSFOM for FPGM are [11, 17]

$$h_{i+1,k} = \begin{cases} \frac{1}{t_{i+1}}\left(t_k - \sum_{j=k+1}^{i} h_{j,k}\right), & k = 0, \ldots, i-1, \\ 1 + \frac{t_i - 1}{t_{i+1}}, & k = i. \end{cases}$$

worst-case rate for decreasing the cost function, using Nesterov's acceleration technique [23, 24].

Algorithm FPGM (FISTA)

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $t_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i)$$

$$t_{i+1} = \frac{1 + \sqrt{1 + 4t_i^2}}{2} \quad (11)$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{x}_{i+1} + \frac{t_i - 1}{t_{i+1}}(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)$$

FPGM has the following bound for the cost function [3, Thm. 4.4] for any $N \geq 1$:

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \leq \frac{LR^2}{2t_{N-1}^2} \leq \frac{2LR^2}{(N+1)^2}, \quad (12)$$

where the parameters t_i (11) satisfy

$$t_i^2 = \sum_{l=0}^{i} t_l \text{ and } t_i \geq \frac{i+2}{2}. \quad (13)$$

Sec. 3 provides a new proof of the cost function bound (12) of FPGM using a new relaxation of PEP, and illustrates that this particular acceleration of PGM results from optimizing a relaxed version of the cost function form of PEP. In addition, it is shown in [3, 5] that FPGM and its bound (12) generalize to any t_i such that $t_0 = 1$ and $t_i^2 \leq t_{i-1}^2 + t_i$ for all $i \geq 1$ with corresponding bound for any $N \geq 1$:

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \leq \frac{LR^2}{2t_{N-1}^2}, \quad (14)$$

which includes the choice $t_i = \frac{i+a}{a}$ for any $a \geq 2$. Using our relaxed PEP, Sec. 3 further describes similar but different generalizations of FPGM that complement our understanding of FPGM.

We are often interested in the worst-case analysis of the norm of the (composite) gradient (mapping) in addition to that of the cost function, particularly when dealing with dual problems. To improve the rate $O(1/N)$ of the gradient norm bound of a gradient method, Nesterov [26] suggested performing his fast gradient method (FGM) [23, 24], a non-proximal version of FPGM, for the first m iterations and a gradient method for remaining $N - m$ iterations for smooth convex problems (when $\phi(x) = 0$). Here we extend this idea to the nonsmooth composite convex problem (M) and use FPGM-m to denote the resulting algorithm. The following theorem provides a $O(1/N^{\frac{3}{2}})$ worst-case bound for the norm of composite gradient mapping of FPGM-m, using the idea in [26] and Lemma 1.

Algorithm FPGM-m

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $y_0 = x_0$, $t_0 = 1$.

For $i = 0, \ldots, N-1$

$$x_{i+1} = \mathbf{p}_L(y_i)$$

$$t_{i+1} = \frac{1 + \sqrt{1 + 4t_i^2}}{2}, \quad i \leq m-1$$

$$y_{i+1} = \begin{cases} x_{i+1} + \frac{t_i - 1}{t_{i+1}}(x_{i+1} - x_i), & i \leq m-1, \\ x_{i+1}, & \text{otherwise.} \end{cases}$$

Theorem 3

Let $F: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $x_0, \cdots, x_N \in \mathbb{R}^d$ be generated by FPGM-m for $1 \leq m \leq N$. Then for $N \geq 1$,

$$\min_{i \in \{0, \ldots, N\}} \|\tilde{\nabla}_L F(x_i)\| \leq \|\tilde{\nabla}_L F(x_N)\| \leq \frac{2LR}{(m+1)\sqrt{N-m+1}}. \quad (15)$$

Proof—We have

$$\frac{2LR^2}{(m+1)^2} \overset{(12)}{\geq} F(\boldsymbol{x}_m) - F(\boldsymbol{x}_*) \overset{(6)}{\geq} F(\boldsymbol{x}_{N+1}) - F(\boldsymbol{x}_*) + \frac{1}{2L} \sum_{i=m}^{N} \| \widetilde{\nabla}_L F(\boldsymbol{x}_i) \|^2 \overset{(9)}{\geq} \frac{N-m+1}{2L} \| \widetilde{\nabla}_L F(\boldsymbol{x}_N) \|^2,$$

which is equivalent to (15).

As noticed by a reviewer, when $m = \lfloor \frac{2N}{3} \rfloor$, the worst-case bound (15) of the composite gradient mapping roughly has its smallest constant $3\sqrt{3}$ for the rate $O(1/N^{\frac{3}{2}})$, which is better than the choice $m = \lfloor \frac{N}{2} \rfloor$ in [26].

Monteiro and Svaiter [21] considered a variant of FPGM that replaces $\mathbf{p}_L(\cdot)$ of FPGM by $\mathbf{p}_{L/\sigma^2}(\cdot)$ for $0 < \sigma < 1$; that variant, which we denote FPGM-σ, satisfies the $O(1/N^{\frac{3}{2}})$ rate for the composite gradient mapping. This FPGM-σ algorithm satisfies the following cost function and composite gradient mapping worst-case bounds[5] [21, Prop. 5.2] for $N \geq 1$:

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \leq \frac{2LR^2}{\sigma^2 N^2}, \quad (16)$$

$$\min_{i \in \{0, \ldots, N\}} \| \widetilde{\nabla}_{L/\sigma^2} F(\boldsymbol{y}_i) \| \leq \frac{2\sqrt{3}}{\sigma} \sqrt{\frac{1+\sigma}{1-\sigma}} \frac{LR}{N^{\frac{3}{2}}}. \quad (17)$$

The worst-case bound (17) of the composite gradient mapping has its smallest constant $\frac{2\sqrt{3}}{\sigma^2}\sqrt{\frac{1+\sigma}{1-\sigma}} \approx 16.2$ when $\sigma = \frac{\sqrt{17}-1}{4} \approx 0.78$, which makes the bound (17) about $\frac{16}{3\sqrt{3}} \approx 3$-times larger than the bound (15) of FPGM$-(m = \lfloor \frac{2N}{3} \rfloor)$ at best. However, since FPGM-σ does not require one to select the number of total iterations N in advance unlike FPGM-m, the FPGM-σ algorithm could be useful in practice, as discussed further in Sec. 4.4. Ghadimi and Lan [13] also showed the $O(1/N^{\frac{3}{2}})$ rate for a composite gradient mapping worst-case bound of another variant of FPGM, but the corresponding algorithm in [13] requires two proximal gradient updates per iteration, combining the proximal operations and the gradient steps in a way that differs from the class FSFOM and could be less attractive in terms of the per-iteration computational complexity.

[5] The bound for $\min_{i \in \{0, \ldots, N\}} \| \widetilde{\nabla}_{L/\sigma^2} F(\boldsymbol{y}_i) \|$ of FPGM-σ is described in a big-O sense in [21, Prop. 5.2(c)], and we further computed the constant in (17) by following the derivation of [21, Prop. 5.2(c)].

FPGM has been used in dual problems [1, 2, 4, 14]; using FPGM-m and the algorithms in [13, 21] that guarantee $O(1/N^{\frac{3}{2}})$ rate for minimizing the norm of the composite gradient mapping could be potentially useful for solving dual problems. (Using F(P)GM-m for (dual) smooth convex problems was discussed in [9, 22, 26].) However, FPGM-m and the algorithms in [13, 21] are not necessarily the best possible methods with respect to the worst-case bound of the norm of the composite gradient mapping. Therefore, Sec. 4 seeks to optimize the step coefficients of FSFOM for minimizing the norm of the composite gradient mapping using a relaxed PEP.

The next section first provides a new proof of FPGM using our new relaxation on PEP, and proposes the new generalized FPGM.

3. Relaxation and optimization of the cost function form of PEP

3.1. Relaxation for the cost function form of PEP

For FSFOM with given step-size coefficients $\boldsymbol{h} \coloneqq \{h_{i+1,k}\}$, in principle the worst-case bound on the cost function after N iterations corresponds to the solution of the following PEP problem [11]:

$$\begin{aligned} \mathcal{B}_{\mathrm{P}}(\boldsymbol{h}, N, d, L, R) \coloneqq \max_{\substack{F \in \mathcal{F}_L(\mathbb{R}^d), \\ \boldsymbol{x}_0, \cdots, \boldsymbol{x}_N \in \mathbb{R}^d, \boldsymbol{x}_* \in X_*(F) \\ \boldsymbol{y}_0, \cdots, \boldsymbol{y}_{N-1} \in \mathbb{R}^d}} & F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \\ \text{s.t. } & \boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i), \quad i = 0, \ldots, N-1, \quad \|\boldsymbol{x}_0 - \boldsymbol{x}_*\| \le R, \\ & \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \sum_{k=0}^{i} h_{i+1,k}(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k), \quad i = 0, \ldots, N-2. \end{aligned} \tag{P}$$

Since (non-relaxed) PEP problems like (P) are difficult to solve due to the (infinite-dimensional) functional constraint on F, Drori and Teboulle [11] suggested (for smooth convex problems) replacing the functional constraint by a property of F related to the update such as $\mathbf{p}_L(\cdot)$ in (P). Taylor et al. [29, 30] discussed properties of F that can replace the functional constraint of PEP without strictly relaxing (P), and provided tight numerical worst-case analysis for any given N. However, analytical solutions remain unknown for (P) and most PEP problems.

Instead, this paper proposes an alternate relaxation that is looser than that in [29, 30] but provides tractable and useful analytical results. We consider the following property of F involving the proximal gradient update $\mathbf{p}_L(\cdot)$ [3, Lemma 2.3]:

$$F(\boldsymbol{x}) - F(\mathbf{p}_L(\boldsymbol{y})) \le \frac{L}{2}\|\mathbf{p}_L(\boldsymbol{y}) - \boldsymbol{y}\|^2 + L\langle \boldsymbol{y} - \boldsymbol{x}, \mathbf{p}_L(\boldsymbol{y}) - \boldsymbol{y}\rangle \quad \forall_{\boldsymbol{x},\boldsymbol{y}} \in \mathbb{R}^d \tag{18}$$

to replace the functional constraint on F. In particular, we use the following property:

$$\frac{L}{2}\|\mathbf{p}_L(y)-y\|^2 - L\langle \mathbf{p}_L(x)-x, \mathbf{p}_L(y)-y\rangle \le F(\mathbf{p}_L(x)) - F(\mathbf{p}_L(y)) + L\langle \mathbf{p}_L(y)-y, x-y\rangle, \quad \forall_{x,y}\in\mathbb{R}^d \tag{19}$$

that results from replacing $\boldsymbol{x}$ in (18) by $\mathbf{p}_L(\boldsymbol{x})$. When $\phi(\boldsymbol{x}) = 0$, the property (19) reduces to

$$\frac{1}{2L}\|\nabla f(y)\|^2 - \frac{1}{L}\langle \nabla f(x), \nabla f(y)\rangle \le f\left(x-\frac{1}{L}\nabla f(x)\right) - f\left(y-\frac{1}{L}\nabla f(y)\right) - \langle \nabla f(y), x-y\rangle, \quad \forall_{x,y}\in\mathbb{R}^d. \tag{20}$$

Note that the relaxation of PEP in [11, 16, 17, 18, 30] for unconstrained smooth convex minimization ($\phi(\boldsymbol{x}) = 0$) uses a well-known property of f in [24, Thm. 2.1.5] that differs from (20) and does not strictly relax the PEP as discussed in [30], whereas our relaxation using (19) and (20) does not guarantee a tight relaxation of (P). Finding a tight relaxation that leads to useful (or even optimal) algorithms remains an open problem for nonsmooth composite convex problems.

Similar to [11, Problem (Q′)], we (strictly) relax problem (P) as follows using a set of constraint inequalities (19) at the points $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{y}_{i-1}, \boldsymbol{y}_i)$ for $i = 1, \ldots, N-1$ and $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}_*, \boldsymbol{y}_i)$ for $i = 0, \ldots, N-1$:

$$\begin{aligned} \mathscr{B}_{\mathrm{P1}}(\boldsymbol{h}, N, d, L, R) \coloneqq \max_{\substack{\boldsymbol{G}\in\mathbb{R}^{N\times d},\\ \boldsymbol{\delta}\in\mathbb{R}^N}} \; & LR^2\delta_{N-1} \\ \mathrm{s.t.}\; & \mathtt{Tr}\{\boldsymbol{G}^\top \breve{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h})\boldsymbol{G}\} \le \delta_{i-1} - \delta_i, \quad i = 1, \ldots, N-1, \\ & \mathtt{Tr}\{\boldsymbol{G}^\top \breve{\boldsymbol{D}}_i(\boldsymbol{h})\boldsymbol{G} + \boldsymbol{\nu}\boldsymbol{u}_i^\top \boldsymbol{G}\} \le -\delta_i, \quad i = 0, \ldots, N-1, \end{aligned} \tag{P1}$$

for any given unit vector $\boldsymbol{\nu}\in\mathbb{R}^d$, by defining the $(i+1)$th standard basis vector $\boldsymbol{u}_i = \boldsymbol{e}_{i+1} \in \mathbb{R}^N$, the matrix $\boldsymbol{G} = [\boldsymbol{g}_0, \cdots, \boldsymbol{g}_{N-1}]^\top \in \mathbb{R}^{N\times d}$ and the vector $\boldsymbol{\delta} = [\delta_0, \cdots, \delta_{N-1}]^\top \in \mathbb{R}^N$, where

$$\begin{cases} \boldsymbol{g}_i \coloneqq -\dfrac{1}{\|\boldsymbol{y}_0 - \boldsymbol{x}_*\|}(\mathbf{p}_L(\boldsymbol{y}_i) - \boldsymbol{y}_i) = \dfrac{1}{L\|\boldsymbol{y}_0 - \boldsymbol{x}_*\|}\widetilde{\nabla}_L F(\boldsymbol{y}_i), \\ \delta_i \coloneqq \dfrac{1}{L\|\boldsymbol{y}_0 - \boldsymbol{x}_*\|^2}(F(\mathbf{p}_L(\boldsymbol{y}_i)) - F(\boldsymbol{x}_*)), \end{cases} \tag{21}$$

for $i = 0, \ldots, N-1, *$. Note that $\boldsymbol{g}_* = [0, \cdots, 0]^\top$, $\delta_* = 0$ and $\mathrm{Tr}\{\boldsymbol{G}^\top \boldsymbol{u}_i \boldsymbol{u}_j^\top \boldsymbol{G}\} = \langle \boldsymbol{g}_i, \boldsymbol{g}_j \rangle$ by definition. The matrices $\check{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h})$ and $\check{\boldsymbol{D}}_i(\boldsymbol{h})$ are defined as

$$\begin{cases} \check{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h}) \coloneqq \frac{1}{2}\boldsymbol{u}_i\boldsymbol{u}_i^\top - \frac{1}{2}\boldsymbol{u}_{i-1}\boldsymbol{u}_i^\top - \frac{1}{2}\boldsymbol{u}_i\boldsymbol{u}_{i-1}^\top + \frac{1}{2}\sum_{k=0}^{i-1} h_{i,k}(\boldsymbol{u}_i\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_i^\top), \\ \check{\boldsymbol{D}}_i(\boldsymbol{h}) \coloneqq \frac{1}{2}\boldsymbol{u}_i\boldsymbol{u}_i^\top + \frac{1}{2}\sum_{j=1}^{i}\sum_{k=0}^{j-1} h_{j,k}(\boldsymbol{u}_i\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_i^\top), \end{cases} \tag{22}$$

which results from the inequalities (19) at the points $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{y}_{i-1}, \boldsymbol{y}_i)$ and $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}_*, \boldsymbol{y}_i)$ respectively.

As in [11, Problem (DQ′)], problem (P1) has a dual formulation that one can solve numerically for any given N using a semidefinite program (SDP) to determine an upper bound on the cost function worst-case bound for any FSFOM:[6]

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \le \mathscr{B}_{\mathrm{P}}(\boldsymbol{h}, N, d, L, R) \le \mathscr{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R) \coloneqq \max_{\substack{(\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda, \\ \gamma \in \mathbb{R}}} \left\{ \frac{1}{2}LR^2\gamma : \begin{pmatrix} \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq 0 \right\}, \tag{D}$$

where $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_{N-1}]^\top \in \mathbb{R}_+^{N-1}$, $\boldsymbol{\tau} = [\tau_0, \cdots, \tau_{N-1}]^\top \in \mathbb{R}_+^N$, and

$$\Lambda = \left\{ (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \mathbb{R}_+^{2N-1} : \begin{matrix} \tau_0 = \lambda_1, \lambda_{N-1} + \tau_{N-1} = 1, \\ \lambda_i - \lambda_{i+1} + \tau_i = 0, i = 1, \ldots, N-2, \end{matrix} \right\}, \tag{23}$$

$$\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = \sum_{i=1}^{N-1} \lambda_i \check{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h}) + \sum_{i=0}^{N-1} \tau_i \check{\boldsymbol{D}}_i(\boldsymbol{h}). \tag{24}$$

[6] See Appendix A for the derivation of the dual formulation (D) of (P1).

This means that one can compute a valid upper bound (D) of (P) for given step coefficients $\boldsymbol{h}$ using a SDP. The next two sections provide an analytical solution to (D) for FPGM and similarly for our new generalized FPGM, superseding the use of numerical SDP solvers.

3.2. Generalized FPGM

We specify a feasible point of (D) that leads to our new generalized form of FPGM.

Lemma 4—For the following step coefficients:

$$h_{i+1,k} = \begin{cases} \frac{t_{i+1}}{T_{i+1}}\left(t_k - \sum_{j=k+1}^{i} h_{j,k}\right), & k = 0, \ldots, i-1, \\ 1 + \frac{(t_i - 1)t_{i+1}}{T_{i+1}}, & k = i, \end{cases} \tag{25}$$

the choice of variables:

$$\lambda_i = \frac{T_{i-1}}{T_{N-1}}, \quad i = 1, \ldots, N-1, \quad \tau_i = \frac{t_i}{T_{N-1}}, \quad i = 0, \ldots, N-1, \quad \gamma = \frac{1}{T_{N-1}}, \tag{26}$$

is a feasible point of (D) for any choice of t_i such that

$$t_0 = 1, \quad t_i > 0, \quad \textit{and} \quad t_i^2 \le T_i := \sum_{l=0}^{i} t_l. \tag{27}$$

Proof: It is obvious that $(\boldsymbol{\lambda}, \boldsymbol{\tau})$ in (26) with (27) is in Λ (23). Using (22), the (i, k)th entry of the symmetric matrix $\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})$ in (24) can be written as

$$S_{i,k}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = \begin{cases} \frac{1}{2}((\lambda_i + \tau_i)h_{i,k} + \tau_i \sum_{j=k+1}^{i-1} h_{j,k}), & i = 2, \ldots, N-1, k = 0, \ldots, i-2, \\ \frac{1}{2}((\lambda_i + \tau_i)h_{i,i-1} - \lambda_i), & i = 1, \ldots, N-1, k = i-1, \\ \frac{1}{2}\lambda_{i+1}, & i = 0, \ldots, N-2, k = i, \\ \frac{1}{2}, & i = N-1, k = i, \end{cases}$$

where each element $S_{i,k}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})$ corresponds to the coefficient of the term $\boldsymbol{u}_i \boldsymbol{u}_k^\top$ of $\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})$ in (24). Then, inserting (25) and (26) to the above yields

$$S_{i,k}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau}) =$$

$$\begin{cases} \frac{1}{2}\left(\frac{T_i}{T_{N-1}}\frac{t_i}{T_i}\left(t_k - \sum_{j=k+1}^{i-1} h_{j,k}\right) + \frac{t_i}{T_{N-1}}\sum_{j=k+1}^{i-1} h_{j,k}\right), & i = 2,\ldots,N-1, k = 0,\ldots,i-2, \\ \frac{1}{2}\left(\frac{T_i}{T_{N-1}}\left(1 + \frac{(t_{i-1}-1)t_i}{T_i}\right) - \frac{T_{i-1}}{T_{N-1}}\right), & i = 1,\ldots,N-1, k = i-1, \\ \frac{T_i}{2T_{N-1}}, & i = 0,\ldots,N-1, k = i. \end{cases} =$$

$$\begin{cases} \frac{t_i t_k}{2T_{N-1}}, & i = 1,\ldots,N-1, k = 0,\ldots,i-1, \\ \frac{T_i}{2T_{N-1}}, & i = 0,\ldots,N-1, k = i. \end{cases}$$

Then, using (26) and (27), we finally show the feasibility condition of (D):

$$\begin{pmatrix} \boldsymbol{S}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} = \frac{1}{2T_{N-1}}(\mathtt{diag}\{\boldsymbol{T} - \boldsymbol{t}^2\} + \boldsymbol{t}\boldsymbol{t}^\top) \succeq 0,$$

where $\boldsymbol{t} = (t_0, \cdots, t_{N-1}, 1)^\top$ and $\boldsymbol{T} = (T_0, \cdots, T_{N-1}, 1)^\top$.

FSFOM with the step coefficients (25) would be both computationally and memory-wise inefficient, so we next present an equivalent recursive form of FSFOM with (25), named Generalized FPGM (GFPGM).

Algorithm GFPGM

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $t_0 = T_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i)$$

$$\text{Choose } t_{i+1} \text{ s.t. } t_{i+1} > 0 \text{ and } t_{i+1}^2 \le T_{i+1} := \sum_{l=0}^{i+1} t_l$$

$$y_{i+1} = x_{i+1} + \frac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}}(x_{i+1} - x_i) + \frac{(t_i^2 - T_i)t_{i+1}}{t_i T_{i+1}}(x_{i+1} - y_i)$$

Proposition 5—The sequence $\{x_0, \cdots, x_N\}$ generated by FSFOM with step sizes (25) is identical to the corresponding sequence generated by GFPGM.

Proof: See Appendix B.

Using Lemma 4, the following theorem bounds the cost function for the GFPGM iterates.

Theorem 6—Let $F: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $x_0, \cdots, x_N \in \mathbb{R}^d$ be generated by GFPGM. Then for $N \geq 1$,

$$F(x_N) - F(x_*) \leq \frac{LR^2}{2T_{N-1}}. \quad (28)$$

Proof: Using (D), Lemma 4 and Prop. 5, we have

$$F(x_N) - F(x_*) \leq \mathcal{B}_{\mathrm{D}}(h, N, L, R) = \frac{1}{2}LR^2\gamma = \frac{LR^2}{2T_{N-1}}. \quad (29)$$

The GFPGM and Thm. 6 reduce to FPGM and (12) when $t_i^2 = T_i$ for all i, and Sec. 3.4 describes that FPGM results from optimizing the step coefficients of FSFOM with respect to the cost function form of the relaxed PEP (D). This GFPGM also includes the choice $t_i = \frac{i+a}{a}$ for any $a \geq 2$ as used in [5], which we denote as FPGM-a that differs from the algorithm in [5]. The following corollary provides a cost function worst-case bound for FPGM-a.

Corollary 7—Let $F: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $x_0, \cdots, x_N \in \mathbb{R}^d$ be generated by GFPGM with $t_i = \frac{i+a}{a}$ (FPGM-a) for any $a \geq 2$. Then for $N \geq 1$,

$$F(x_N) - F(x_*) \leq \frac{aLR^2}{N(N+2a-1)}. \quad (30)$$

Proof: Thm. 6 implies (30), since $t_i = \frac{i+a}{a}$ satisfies (27), *i.e.*,

$$T_i - t_i^2 = \frac{(i+1)(i+2a)}{2a} - \frac{(i+a)^2}{a^2} = \frac{(a-2)i^2 + a(2a-3)i}{2a^2} \geq 0 \quad (31)$$

for any $a \geq 2$ and all $i \geq 0$.

3.3. Related work of GFPGM

This section shows that the GFPGM has a close connection to the accelerated algorithm in [25] that was developed specifically for a constrained smooth convex problem with a closed convex set Q, *i.e.*,

$$\phi(\boldsymbol{x}) = \mathrm{I}_Q(\boldsymbol{x}) \coloneqq \begin{cases} 0, & \boldsymbol{x} \in Q, \\ \infty, & \text{otherwise.} \end{cases} \quad (32)$$

The projection operator $\mathrm{P}_Q(\boldsymbol{x}) \coloneqq \arg\min_{\boldsymbol{y} \in Q} \|\boldsymbol{x} - \boldsymbol{y}\|$ is used for the proximal gradient update (2).

We show that the GFPGM can be written in the following equivalent form, named GFPGM′, which is similar to that of the accelerated algorithm in [25] shown below. Note that the accelerated algorithm in [25] satisfies the bound (28) of the GFPGM in [25, Thm. 2] when $\phi(\boldsymbol{x}) = \mathrm{I}_Q(\boldsymbol{x})$.

Algorithm GFPGM′

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $t_0 = T_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i) = \boldsymbol{y}_i - \frac{1}{L}\widetilde{\nabla}_L F(\boldsymbol{y}_i)$$

$$\boldsymbol{z}_{i+1} = \boldsymbol{y}_0 - \frac{1}{L}\sum_{k=0}^{i} t_k \widetilde{\nabla}_L F(\boldsymbol{y}_k)$$

Choose t_{i+1} s.t. $t_{i+1} > 0$ and $t_{i+1}^2 \leq T_{i+1} = \sum_{l=0}^{i+1} t_l$

$$\boldsymbol{y}_{i+1} = \left(1 - \frac{t_{i+1}}{T_{i+1}}\right)\boldsymbol{x}_{i+1} + \frac{t_{i+1}}{T_{i+1}}\boldsymbol{z}_{i+1}$$

Algorithm [25] **for** $\phi(\boldsymbol{x}) = \mathrm{I}_Q(\boldsymbol{x})$

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $t_0 = T_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i) = \mathrm{P}_Q\left(\boldsymbol{y}_i - \frac{1}{L}\nabla f(\boldsymbol{y}_i)\right)$$

$$z_{i+1} = \mathrm{P}_Q\left(\boldsymbol{y}_0 - \frac{1}{L}\sum_{k=0}^{i} t_k \nabla f(\boldsymbol{y}_k)\right)$$

Choose t_{i+1} s.t. $t_{i+1} > 0$ and $t_{i+1}^2 \le T_{i+1} = \sum_{l=0}^{i+1} t_l$

$$\boldsymbol{y}_{i+1} = \left(1 - \frac{t_{i+1}}{T_{i+1}}\right)\boldsymbol{x}_{i+1} + \frac{t_{i+1}}{T_{i+1}} z_{i+1}$$

Proposition 8—The sequence $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N\}$ generated by GFPGM is identical to the corresponding sequence generated by GFPGM′.

Proof: See Appendix C.

Clearly GFPGM′ and the accelerated algorithm in [25] are equivalent for the unconstrained smooth convex problem ($Q = \mathbb{R}^d$). However, when the operation $\mathrm{P}_Q(\boldsymbol{x})$ is relatively expensive, our GFPGM and GFPGM′ that use one projection per iteration could be preferred over the accelerated algorithm in [25] that uses two projections per iteration.

3.4. Optimizing step coefficients of FSFOM using the cost function form of PEP

To find the step coefficients in the class FSFOM that are optimal in terms of the cost function form of PEP, we would like to solve the following problem:

$$\widehat{\boldsymbol{h}}_{\mathrm{P}} := \underset{\boldsymbol{h} \in \mathbb{R}^{N(N+1)/2}}{\arg\min} \; \mathscr{B}_{\mathrm{P}}(\boldsymbol{h}, N, d, L, R). \quad \text{(HP)}$$

Because (HP) seems intractable, we instead optimize the step coefficients using the relaxed bound in (D):

$$\widehat{\boldsymbol{h}}_{\mathrm{D}} := \underset{\boldsymbol{h} \in \mathbb{R}^{N(N+1)/2}}{\arg\min} \; \mathscr{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R). \quad \text{(HD)}$$

The problem (HD) is bilinear, and a convex relaxation technique in [11, Thm. 3] makes it solvable using numerical methods. We optimized (HD) numerically for many choices of N using a SDP solver [7, 15] and based on our numerical results (not shown) we conjecture that the feasible point in Lemma 4 with $t_i^2 = T_i$ that corresponds to FPGM (FISTA) is a global minimizer of (HD). It is straightforward to show that the step coefficients in Lemma 4 with $t_i^2 = T_i$ give the smallest bound of (D) and (28) among all feasible points in Lemma 4, but showing optimality among all possible feasible points of (HD) may require further derivations as in [17, Lemma 3] using KKT conditions, which we leave as future work.

This section has provided a new worst-case bound proof of FPGM using the relaxed PEP, and suggested that FPGM corresponds to FSFOM with optimized step coefficients using the cost function form of the relaxed PEP. The next section provides a different optimization of the step coefficients of FSFOM that targets the norm of the composite gradient mapping, because minimizing the norm of the composite gradient mapping is important in dual problems (see [9, 22, 26] and (8)).

4. Relaxation and optimization of the composite gradient mapping form of PEP

4.1. Relaxation for the composite gradient mapping form of PEP

To form a worst-case bound on the norm of the composite gradient mapping for a given $\boldsymbol{h}$ of FSFOM, we use the following PEP that replaces $F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*)$ in (P) by the norm squared of the composite gradient mapping. Here, we consider the smallest composite gradient mapping norm squared among all iterates[7] ($\min_{\boldsymbol{x} \in \Omega_N} \|L(\mathbf{p}_L(\boldsymbol{x}) - \boldsymbol{x})\|^2 = \min_{\boldsymbol{x} \in \Omega_N} \|\tilde{\nabla}_L F(\boldsymbol{x})\|^2$ where $\Omega_N := \{\boldsymbol{y}_0, \cdots, \boldsymbol{y}_{N-1}, \boldsymbol{x}_N\}$) as follows:

$$
\begin{aligned}
\mathcal{B}_{\mathrm{P}'}(\boldsymbol{h}, N, d, L, R) := \quad & \max_{F \in \mathcal{F}_L(\mathbb{R}^d),} \quad \min_{\boldsymbol{x} \in \Omega_N} \|L(\mathbf{p}_L(\boldsymbol{x}) - \boldsymbol{x})\|^2 \\
& \boldsymbol{x}_0, \cdots, \boldsymbol{x}_N \in \mathbb{R}^d, \boldsymbol{x}_* \in X_*(F), \\
& \boldsymbol{y}_0, \cdots, \boldsymbol{y}_{N-1} \in \mathbb{R}^d \qquad (\mathrm{P}') \\
& \text{s.t. } \boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i), \quad i = 0, \ldots, N-1, \quad \|\boldsymbol{x}_0 - \boldsymbol{x}_*\| \le R, \\
& \boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \sum_{k=0}^{i} h_{i+1,k}(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k), \quad i = 0, \ldots, N-2.
\end{aligned}
$$

Because this infinite-dimensional max-min problem appears intractable, similar to the relaxation from (P) to (P1), we relax (P') to a finite-dimensional problem with an additional constraint resulting from (6) that is equivalent to

[7] See Appendix D for the discussion on the choice of Ω_N.

$$\frac{L}{2}\|\mathbf{p}_L(\boldsymbol{x}_N)-\boldsymbol{x}_N\|^2 \le F(\boldsymbol{x}_N)-F(\boldsymbol{x}_*) \quad (33)$$

and conditions that are equivalent to $\alpha \le \|L(\mathbf{p}_L(\boldsymbol{x})-\boldsymbol{x})\|^2$ for all $\boldsymbol{x}\in\Omega_N$ after replacing $\min_{\boldsymbol{x}\in\Omega_N}\|L(\mathbf{p}_L(\boldsymbol{x})-\boldsymbol{x})\|^2$ by α as in [30].[8] This relaxation leads to

$$\begin{aligned}\mathcal{B}_{\mathrm{P1}'}(\boldsymbol{h},N,d,L,R) \coloneqq \max_{\substack{\bar{\boldsymbol{G}}\in\mathbb{R}^{(N+1)\times d},\\ \boldsymbol{\delta}\in\mathbb{R}^N,\alpha\in\mathbb{R}}} & L^2R^2\alpha \\ \text{s.t. } & \mathtt{Tr}\{\bar{\boldsymbol{G}}^\top \bar{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h})\bar{\boldsymbol{G}}\} \le \delta_{i-1}-\delta_i, \quad i=1,\dots,N-1,\\ & \mathtt{Tr}\{\bar{\boldsymbol{G}}^\top \bar{\boldsymbol{D}}_i(\boldsymbol{h})\bar{\boldsymbol{G}}+\boldsymbol{\nu}\bar{\boldsymbol{u}}_i^\top\bar{\boldsymbol{G}}\} \le -\delta_i, \quad i=0,\dots,N-1,\\ & \mathtt{Tr}\left\{\frac{1}{2}\bar{\boldsymbol{G}}^\top\bar{\boldsymbol{u}}_N\bar{\boldsymbol{u}}_N^\top\bar{\boldsymbol{G}}\right\} \le \delta_{N-1},\\ & \mathtt{Tr}\{-\bar{\boldsymbol{G}}^\top\bar{\boldsymbol{u}}_i\bar{\boldsymbol{u}}_i^\top\bar{\boldsymbol{G}}\} \le -\alpha, \quad i=0,\dots,N,\end{aligned} \quad (\mathrm{P1}')$$

for any given unit vector $\boldsymbol{\nu}\in\mathbb{R}^d$, by defining the $(i+1)$th standard basis vector $\bar{\boldsymbol{u}}_i = \boldsymbol{e}_{i+1} \in \mathbb{R}^{N+1}$, the matrices

$$\bar{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h}) = \begin{pmatrix}\check{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h}) & \mathbf{0}\\ \mathbf{0}^\top & 0\end{pmatrix}, \quad \bar{\boldsymbol{D}}_i(\boldsymbol{h}) = \begin{pmatrix}\check{\boldsymbol{D}}_i(\boldsymbol{h}) & \mathbf{0}\\ \mathbf{0}^\top & 0\end{pmatrix} \quad (34)$$

where $\mathbf{0} = [0,\dots,0]^\top \in \mathbb{R}^N$, and the matrix $\bar{\boldsymbol{G}} = [\boldsymbol{G}^\top, \bar{\boldsymbol{g}}_N]^\top \in \mathbb{R}^{(N+1)\times d}$ where

$$\bar{\boldsymbol{g}}_N \coloneqq -\frac{1}{\|\boldsymbol{y}_0-\boldsymbol{x}_*\|}(\mathbf{p}_L(\boldsymbol{x}_N)-\boldsymbol{x}_N) = \frac{1}{L\|\boldsymbol{y}_0-\boldsymbol{x}_*\|}\tilde{\nabla}_L F(\boldsymbol{x}_N). \quad (35)$$

Similar to (D) and [16, Problem (D″)], we have the following dual formulation of (P1′) that could be solved using SDP:

$$\mathcal{B}_{\mathrm{D}'}(\boldsymbol{h},N,L,R) \coloneqq \min_{\substack{(\lambda,\boldsymbol{\tau},\eta,\boldsymbol{\beta})\in\Lambda',\\ \gamma\in\mathbb{R}}} \left\{\frac{1}{2}L^2R^2\gamma : \begin{pmatrix}\boldsymbol{S}'(\boldsymbol{h},\lambda,\boldsymbol{\tau},\eta,\boldsymbol{\beta}) & \frac{1}{2}[\boldsymbol{\tau}^\top,0]^\top\\ \frac{1}{2}[\boldsymbol{\tau}^\top,0] & \frac{1}{2}\gamma\end{pmatrix} \succeq 0\right\} \quad (\mathrm{D}')$$

where $\eta\in\mathbb{R}_+$, $\boldsymbol{\beta} = [\beta_0,\cdots,\beta_N]^\top \in \mathbb{R}_+^{N+1}$, and

[8] Here, we simply relaxed (P′) into (P1′) in a way that is similar to the relaxation from (P) to (P1). This relaxation resulted in a constructive analytical worst-case analysis on the composite gradient mapping in this section that is somewhat similar to that on the cost function in Section 3. However, this relaxation on (P1′) turned out to be relatively loose compared to the relaxation on (P1) (see Sec. 5), suggesting there is room for improvement in the future with a tighter relaxation.

$$\Lambda' = \left\{ (\boldsymbol{\lambda}, \boldsymbol{\tau}, \eta, \boldsymbol{\beta}) \in \mathbb{R}_+^{3N+1} : \begin{array}{l} \tau_0 = \lambda_1, \;\; \lambda_{N-1} + \tau_{N-1} = \eta, \;\; \sum_{i=0}^{N} \beta_i = 1, \\ \lambda_i - \lambda_{i+1} + \tau_i = 0, i = 1, \ldots, N-2 \end{array} \right\}, \quad (36)$$

$$\boldsymbol{S}'(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}, \eta, \boldsymbol{\beta}) = \sum_{i=1}^{N-1} \lambda_i \overline{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h}) + \sum_{i=0}^{N-1} \tau_i \overline{\boldsymbol{D}}_i(\boldsymbol{h}) + \frac{1}{2}\eta \bar{\boldsymbol{u}}_N \bar{\boldsymbol{u}}_N^\top - \sum_{i=0}^{N} \beta_i \bar{\boldsymbol{u}}_i \bar{\boldsymbol{u}}_i^\top . \quad (37)$$

The next section specifies a feasible point of interest that is in the class of GFPGM and analyzes the worst-case bound of the norm of the composite gradient mapping. Then we optimize the step coefficients of FSFOM with respect to the composite gradient mapping form of PEP leading to a new algorithm that differs from Nesterov's acceleration for decreasing the cost function.

4.2. Worst-case analysis of the composite gradient mapping of GF-PGM

The following lemma provides feasible point of (D′) for the step coefficients (25) of GFPGM.

Lemma 9—For the step coefficients $\{h_{i+1,k}\}$ in (25), the choice of variables

$$\lambda_i = T_{i-1}\tau_0, \;\; i = 1, \ldots, N-1, \;\; \tau_i = \begin{cases} \left(\frac{1}{2} \left(\sum_{k=0}^{N-1} (T_k - t_k^2) + T_{N-1} \right) \right)^{-1}, & i = 0, \\ t_i \tau_0, & i = 1, \ldots, N-1, \end{cases}$$

(38)

$$\eta = T_{N-1}\tau_0, \;\; \beta_i = \begin{cases} \frac{1}{2}(T_i - t_i^2)\tau_0, & i = 0, \ldots, N-1, \\ \frac{1}{2} T_{N-1} \tau_0, & i = N, \end{cases} \quad \gamma = \tau_0 . \quad (39)$$

is a feasible point of (D′) for any choice of t_i and T_i satisfying (27).

Proof: It is obvious that $(\boldsymbol{\lambda}, \boldsymbol{\tau}, \eta, \boldsymbol{\beta})$ in (38) and (39) with (27) is in Λ' (36). Using (22) and (34), the (i, k)th entry of the symmetric matrix $\boldsymbol{S}'(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}, \eta, \boldsymbol{\beta})$ in (37) can be written as

$$S'_{i,k}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau},\eta,\boldsymbol{\beta}) = \begin{cases} \frac{1}{2}\left((\lambda_i+\tau_i)h_{i,k} + \tau_i\sum_{j=k+1}^{i-1} h_{j,k}\right), & i=2,\dots,N-1, k=0,\dots,i-2, \\ \frac{1}{2}((\lambda_i+\tau_i)h_{i,i-1} - \lambda_i), & i=1,\dots,N-1, k=i-1, \\ \frac{1}{2}\lambda_{i+1} - \beta_i, & i=0,\dots,N-2, k=i, \\ \frac{1}{2}\eta - \beta_i, & i=N-1,N, k=i, \\ 0, & i=N, k=0,\dots,i-1, \end{cases}$$

and inserting (25), (38), and (39) yields

$$S'_{i,k}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau},\eta,\boldsymbol{\beta}) = \begin{cases} \frac{1}{2}\left(T_i\tau_0\frac{t_i}{T_i}\left(t_k - \sum_{j=k+1}^{i-1} h_{j,k}\right) + t_i\tau_0\sum_{j=k+1}^{i-1} h_{j,k}\right), & i=2,\dots,N-1, k=0,\dots,i-2, \\ \frac{1}{2}\left(T_i\tau_0\left(1+\frac{(t_{i-1}-1)t_i}{T_i}\right) - T_{i-1}\tau_0\right), & i=1,\dots,N-1, k=i-1, \\ \frac{1}{2}T_i\tau_0 - \frac{1}{2}(T_i - t_i^2)\tau_0, & i=0,\dots,N-1, k=i, \\ 0, & i=N, k=0,\dots,i, \end{cases}$$

$$= \begin{cases} \frac{1}{2}t_i t_k \tau_0, & i=0,\dots,N-1, k=0,\dots,i, \\ 0, & i=N, k=0,\dots,i. \end{cases}$$

Finally, by defining $\bar{\boldsymbol{t}} = (t_0, \cdots, t_{N-1}, 0, 1)^\top$ we have the feasibility condition of (D′):

$$\begin{pmatrix} \boldsymbol{S}'(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau},\eta,\boldsymbol{\beta}) & \frac{1}{2}[\boldsymbol{\tau}^\top, 0]^\top \\ \frac{1}{2}[\boldsymbol{\tau}^\top, 0] & \frac{1}{2}\gamma \end{pmatrix} = \frac{1}{2}\bar{t}\bar{t}\tau_0 \succeq 0.$$

Using Lemma 9, the following theorem bounds the (smallest) norm of the composite gradient mapping for the GFPGM iterates.

Theorem 10—Let $f: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N, \mathbf{y}_0, \cdots, \mathbf{y}_{N-1} \in \mathbb{R}^d$ be generated by GFPGM. Then for $N \geq 1$,

$$\min_{i \in \{0,\dots,N\}} \|\widetilde{\nabla}_L F(\boldsymbol{x}_i)\| \leq \min_{\boldsymbol{x} \in \Omega_N} \|\widetilde{\nabla}_L F(\boldsymbol{x})\| \leq \frac{LR}{\sqrt{\sum_{k=0}^{N-1}(T_k - t_k^2) + T_{N-1}}}. \quad (40)$$

Proof: Lemma 1 implies the first inequality of (40). Using (D′), Lemma 9 and Prop. 5, we have

$$\min_{\boldsymbol{x} \in \Omega_N} \|\widetilde{\nabla}_L F(\boldsymbol{x})\|^2 \leq \mathscr{B}_{\mathrm{D}'}(\boldsymbol{h}, N, L, R) = \frac{1}{2} L^2 R^2 \gamma = \frac{L^2 R^2}{\sum_{k=0}^{N-1} (T_k - t_k^2) + T_{N-1}},$$

which is equivalent to (40).

Although the bound (40) is not tight due to the relaxation on PEP, next two sections show that there exists choices of t_i that provide a rate $O(1/N^{\frac{3}{2}})$ for decreasing the composite gradient mapping, including the choice that optimizes the composite gradient mapping form of PEP.

FGM for smooth convex minimization was shown to achieve the rate $O(1/N^{\frac{3}{2}})$ for the decrease of the usual gradient in [16]. In contrast, Thm. 10 provides only a $O(1/N)$ bound for FPGM (or GFPGM with t_i (11)) on the decrease of the composite gradient mapping since $T_i = t_i^2$ for all i and the value of T_{N-1} is $O(N^2)$ for t_i (11). Sec. 5 below numerically studies a tight bound on the composite gradient mapping of FPGM and illustrates that it has a rate that is faster than the rate $O(1/N)$ of Thm. 10, indicating there is a room for improvement in the composite gradient mapping form of the relaxed PEP

4.3. Optimizing step coefficients of FSFOM using the composite gradient mapping form of PEP

To optimize the step coefficients in the class FSFOM in terms of the composite gradient mapping form of the relaxed PEP (D′), we would like to solve the following problem:

$$\widehat{\boldsymbol{h}}_{\mathrm{D}'} := \operatorname*{arg\,min}_{\boldsymbol{h} \in \mathbb{R}^{N(N+1)/2}} \mathscr{B}_{\mathrm{D}'}(\boldsymbol{h}, N, L, R). \quad (\mathrm{HD}')$$

Similar to (HD), we use a convex relaxation [11, Thm. 3] to make the bilinear problem (HD′) solvable using numerical methods. We then numerically optimized (HD′) for many choices of N using a SDP solver [7, 15] and found that the following choice of t_i:

$$t_i = \begin{cases} 1, & i = 0, \\ \dfrac{1 + \sqrt{1 + 4t_{i-1}^2}}{2}, & i = 1, \ldots, \lfloor \frac{N}{2} \rfloor - 1, \\ \dfrac{N - i + 1}{2}, & i = \lfloor \frac{N}{2} \rfloor, \ldots, N - 1, \end{cases} \quad (41)$$

makes the feasible point in Lemma 9 optimal empirically with respect to the relaxed bound (HD′). Interestingly, whereas the usual t_i factors (such as (11) and $t_i = \frac{i+a}{a}$ for any $a \geq 2$) increase with i indefinitely, here, the factors begin decreasing after $i = \lfloor \frac{N}{2} \rfloor - 1$.

We also noticed numerically that finding the t_i that minimizes the bound (40), *i.e.*, solving the following constrained quadratic problem:

$$\max_{\{t_i\}} \left\{ \sum_{k=0}^{N-1} \left(\sum_{l=0}^{k} t_l - t_k^2 \right) + \sum_{l=0}^{N-1} t_l \right\} \quad \text{s.t.} \quad t_i \text{ satisfies (27) for all } i, \qquad (42)$$

is equivalent to optimizing (HD′). This means that the solution of (42) numerically appears equivalent to (41), the (conjectured) solution of (HD′). Interestingly, the unconstrained maximizer of (42) without the constraint (27) is $t_i = \frac{N-i+1}{2}$, and this partially appears in the constrained maximizer (41) of the problem (42).

Based on this numerical evidence, we conjecture that the solution $\hat{\boldsymbol{h}}_{\mathrm{D}'}$ of problem (HD′) corresponds to (25) with (41). Using Prop. 5, the following GFPGM form with (41) is equivalent to FSFOM with the step coefficients (25) for (41) that are optimized step coefficients of FSFOM with respect to the norm of the composite gradient mapping, which we name FPGM-OCG (OCG for optimized over composite gradient mapping).

Algorithm FPGM-OCG (GFPGM with t_i in (41))

Input: $f \in C_L^{1,1}(\mathbb{R}^d)$ convex, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $t_0 = T_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \mathbf{p}_L(\boldsymbol{y}_i)$$

$$t_{i+1} = \begin{cases} \dfrac{1+\sqrt{1+4t_i^2}}{2}, & i = 1, \ldots, \lfloor \frac{N}{2} \rfloor - 2, \\ \dfrac{N-i}{2}, & i = \lfloor \frac{N}{2} \rfloor - 1, \ldots, N-2, \end{cases}$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{x}_{i+1} + \frac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}}(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) + \frac{(t_i^2 - T_i)t_{i+1}}{t_i T_{i+1}}(\boldsymbol{x}_{i+1} - \boldsymbol{y}_i), \quad i < N = 1$$

The following theorem bounds the cost function and the (smallest) norm of the composite gradient mapping for the FPGM-OCG iterates.

Theorem 11—Let $F: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N, \boldsymbol{y}_0, \cdots, \boldsymbol{y}_{N-1} \in \mathbb{R}^d$ be generated by FPGM-OCG. Then for $N \geq 1$,

$$F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*) \leq \frac{4L\|\boldsymbol{x}_0 - \boldsymbol{x}_*\|^2}{N(N+4)}, \quad (43)$$

and for $N \geq 3$,

$$\min_{i \in \{0,\ldots,N\}} \|\widetilde{\nabla}_L F(\boldsymbol{x}_i)\| \leq \min_{\boldsymbol{x} \in \Omega_N} \|\widetilde{\nabla}_L F(\boldsymbol{x})\| \leq \frac{2\sqrt{6}LR}{N\sqrt{N-2}}. \quad (44)$$

Proof: FPGM-OCG is an instance of the GFPGM, and thus Thm. 6 implies (43) using

$$T_{N-1} = T_{m-1} + \sum_{k=m}^{N-1} t_k = t_{m-1}^2 + \sum_{k=m}^{N-1} \frac{N-k+1}{2} = t_{m-1}^2 + \sum_{k'=2}^{N-m+1} \frac{k'}{2} \geq \frac{(m+1)^2}{4} + \frac{(N-m+1)(N-m+2)}{4} - \frac{1}{2} \geq \frac{2N^2+8N+1}{16},$$

where $m = \lfloor \frac{N}{2} \rfloor \geq \frac{N-1}{2}, N-m \geq \frac{N}{2}$, and $T_{m-1} = t_{m-1}^2 \geq \frac{(m+1)^2}{4}$ (13).

In addition, Thm. 10 implies (44), using

$$\sum_{k=0}^{N-1} (T_k - t_k^2) + T_{N-1} \geq \frac{1}{24}(N-2)N^2, \quad (45)$$

which we prove in the Appendix E.

The composite gradient mapping bound (44) of FPGM-OCG is asymptotically $\frac{2\sqrt{2}}{3}$-times smaller than the bound (15) of FPGM$-(m = \lfloor \frac{2N}{3} \rfloor)$. In addition, the cost function bound (43) of FPGM-OCG satisfies the optimal rate $O(1/N^2)$, although the bound (43) is two-times larger than the analogous bound (12) of FPGM.

4.4. Decreasing the composite gradient mapping with a rate $O(1/N^{\frac{3}{2}})$ without selecting N in advance

FPGM-OCG and FPGM-m satisfy a fast rate $O(1/N^{\frac{3}{2}})$ for decreasing the norm of the composite gradient mapping but require one to select the total number of iterations N in advance, which could be undesirable in practice. One could use FPGM-σ in [21] that does not require selecting N in advance, but instead we suggest a new choice of t_i in GFPGM that satisfies a composite gradient mapping bound that is lower than the bound (17) of FPGM-σ.

Based on Thm. 10, the following corollary shows that GFPGM with $t_i = \frac{i+a}{a}$ (FPGM-a) for any $a > 2$ satisfies the rate $O(1/N^{\frac{3}{2}})$ of the norm of the composite gradient mapping without selecting N in advance. (Cor. 7 showed that FPGM-a for any $a \geq 2$ satisfies the optimal rate $O(1/N^2)$ of the cost function.)

Corollary 12—Let $f: \mathbb{R}^d \to \mathbb{R}$ be in $\mathcal{F}^L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_N, \boldsymbol{y}_0, \cdots, \boldsymbol{y}_{N-1} \in \mathbb{R}^d$ be generated by GFPGM with $t_i = \frac{i+a}{a}$ (FPGM-a) for any $a \geq 2$. Then for $N \geq 1$, we have the following bound on the (smallest) composite gradient mapping:

$$\min_{i \in \{0, \ldots, N\}} \|\tilde{\nabla}_L F(\boldsymbol{x}_i)\| \leq \min_{\boldsymbol{x} \in \Omega_N} \|\tilde{\nabla}_L F(\boldsymbol{x})\| \leq \frac{a\sqrt{6}LR}{\sqrt{N((a-2)N^2 + 3(a^2-a+1)N + (3a^2+2a-1))}}. \tag{46}$$

Proof: With $T_i = \frac{(i+1)(i+2a)}{2a}$ and (31), Thm. 10 implies (46) using

$$\sum_{k=0}^{N-1} (T_k - t_k^2) + T_{N-1} = \sum_{k=0}^{N-1} \left(\frac{(k+1)(k+2a)}{2a} - \frac{(k+a)^2}{a^2} \right) + \frac{N(N+2a-1)}{2a} = \sum_{k=0}^{N-1} \left(\frac{(a-2)k^2 + a(2a-3)k}{2a^2} \right) + \frac{N(N+2a-1)}{2a} = \frac{N}{2a^2} \left(\frac{(a-2)(N-1)(2N-1)}{6} + \frac{a(2a-3)(N-1)}{2} + a(N+2a-1) \right)$$

$$= \frac{N((a-2)N^2 + 3(a^2-a+1)N + (3a^2+2a-1))}{6a^2}.$$

FPGM-a for any $a > 2$ has a composite gradient mapping bound (46) that is asymptotically $\frac{a}{2\sqrt{a-2}}$-times larger than the bound (44) of FPGM-OCG. This gap reduces to $\sqrt{2}$ at best when $a = 4$, which is clearly better than that of FPGM-σ. Therefore, this FPGM-a algorithm will be useful for minimizing the composite gradient mapping with a rate $O(1/N^{\frac{3}{2}})$ without selecting N in advance.

5. Discussion

5.1. Summary of analytical worst-case bounds on the cost function and the composite gradient mapping

Table 1 summarizes the *asymptotic* worst-case bounds of all algorithms discussed in this paper. (Note that the bounds are not guaranteed to be tight.) In Table 1, FPGM and FPGM-OCG provide the best known analytical worst-case bounds for decreasing the cost function and the composite gradient mapping respectively. When one does not want to select N in advance for decreasing the composite gradient mapping, FPGM-a will be a useful alternative to FPGM-OCG.

5.2. Tight worst-case bounds on the cost function and the smallest composite gradient mapping norm

Since none of the bounds presented in Table 1 are guaranteed to be tight, we modified the code[9] (using SDP solvers [20, 28]) in Taylor *et al.* [29] to compare tight (numerical) bounds for the cost function and the composite gradient mapping in Tables 2 and 3 respectively for $N = 1, 2, 4, 10, 20, 30, 40, 47, 50$. This numerical bound is guaranteed to be tight when the large-scale condition is satisfied [29]. Taylor *et al.* [29, Fig. 1] already studied a tight worst-case bound on the cost function decrease of FPGM numerically, and found that the analytical bound (12) is asymptotically tight. Table 2 additionally provides numerical tight bounds on the cost function of all algorithms presented in this paper, also suggesting that our relaxation of the cost function form of the PEP from (P) to (D) is asymptotically tight (for some algorithms). In addition, the trend of the tight bounds of the composite gradient mapping in Table 3 follows that of the bounds in Table 1. However, there is gap between them that is not asymptotically tight, unlike the gap between the bounds of the cost function in Tables 1 and 2. In particular, the numerical tight bound for the composite gradient mapping of FPGM in Table 3 has a rate faster than the known rate $O(1/N)$ in Thm. 10. We leave reducing this gap for the bounds on the norm of the composite gradient mapping as future work, possibly with a tighter relaxation of PEP. In addition, FPGM$-(m = \lfloor \frac{2N}{3} \rfloor)$ has a numerical tight bound in Table 3 that is even slightly better than that of FPGM-OCG, unlike our expectation from the analytical bounds in Table 1 and Sec. 4.3. This shows room for improvement in optimizing the step coefficients of FSFOM with respect to the composite gradient mapping, again possibly with a tighter relaxation of PEP.

5.3. Tight worst-case bounds on the final compo site gradient mapping

This paper focused on analyzing the worst-case bound of the *smallest* composite gradient mapping among all iterates ($\min_{\boldsymbol{x}\in\Omega_N} \|\tilde{\nabla}_L F(\boldsymbol{x})\|$) in addition to the cost function, whereas the composite gradient mapping at the *final* iterate ($\|\tilde{\nabla}_L F(\boldsymbol{x}_N)\|$) could be also considered (see Appendix D). For example, the composite gradient mapping bounds (10) and (15) for PGM and FPGM-m also apply to the final composite gradient mapping, and using (6) we can easily derive a (loose) worst-case bound on the final composite gradient mapping for other algorithms, *e.g.*, such a final composite gradient mapping bound for GFPGM is as follows:

$$\|\tilde{\nabla}_L F(\boldsymbol{x}_N)\| \overset{(6)}{\leq} \sqrt{2L(F(\boldsymbol{x}_N) - F(\boldsymbol{p}_L(\boldsymbol{x}_N)))} \leq \sqrt{2L(F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*))} \overset{(28)}{\leq} \frac{LR}{\sqrt{T_{N-1}}}. \quad (47)$$

Since the optimal rate for decreasing the cost function is $O(1/N^2)$, the composite gradient mapping worst-case bound (47) can provide only a rate $O(1/N)$ at best. For completeness of the discussion, Table 4 reports tight numerical bounds for the *final* composite gradient mapping. Here, FPGM, FPGM-($\sigma = 0.78$), and FPGM-($a = 4$) have empirical rates of the

[9]The code in Taylor *et al.* [29] currently does not provide a tight bound of the norm of the composite gradient mapping (and the subgradient), so we simply added a few lines to compute a tight bound.

worst-case bounds in Table 4 that are slower than those in Table 3, unlike the other three including FPGM-OCG.

To best of our knowledge, FPGM-m (or algorithms that similarly perform accelerated algorithms in the beginning and run a proximal gradient method for the remaining iterations) is known only to have a rate $O(1/N^{\frac{3}{2}})$ in (15) for decreasing the final composite gradient mapping, while FPGM-OCG was also found to inherit such fast rate in Table 4. Therefore, searching for first-order methods that have a worst-case bound on the final composite gradient mapping that is lower than that of FPGM-m (and FPGM-OCG), and that possibly do not require knowing N in advance is an interesting open problem. Note that a regularization technique in [26] that provides a faster rate $O(1/N^2)$ (up to a logarithmic factor) for decreasing the final gradient norm for smooth convex minimization can be easily extended for rapidly minimizing the final composite gradient mapping with such rate for the composite problem (M); however, that approach requires knowing R in advance.

5.4. Tight worst-case bounds on the final subgradient

This paper has mainly focused on the norm of the composite gradient mapping based on (8), instead of the subgradient norm that is of primary interest in the dual problem (see *e.g.*, [9, 22, 26]). Therefore to have a better sense of subgradient norm bounds, we computed tight numerical bounds on the *final*[10] subgradient norm $\|F'(\boldsymbol{x}_N)\|$ in Table 5 and compared them with Table 4.

For all six algorithms, empirical rates in Table 5 are similar to those for the final composite gradient mapping in Table 4. In particular, the subgradient norm bounds for the three algorithms PGM, FPGM$-(m = \lfloor \frac{2N}{3} \rfloor)$, and FPGM-OCG are almost identical to those in Table 4 except for the first few iterations, eliminating the concern of using (8) for such cases. On the other hand, the other three algorithms FPGM, FPGM-($\sigma = 0.78$), and FPGM-($a = 4$) almost tightly satisfy the inequality (8) for most N, and thus have bounds on the final subgradient that are about twice larger than those on the final composite gradient mapping. Therefore, regardless of (8), Table 5 further supports the use of FPGM$-(m = \lfloor \frac{2N}{3} \rfloor)$ and FPGM-OCG over FPGM and other algorithms in dual problems.

6. Conclusion

This paper analyzed and developed fixed-step first-order methods (FSFOM) for nonsmooth composite convex cost functions. We showed an alternate proof of FPGM (FISTA) using PEP, and suggested that FPGM (FISTA) results from optimizing the step coefficients of FSFOM with respect to the cost function form of the (relaxed) PEP. We then described a new generalized version of FPGM and analyzed its worst-case bound using the (relaxed)

[10]Using modifications of the code in [29] to compute tight bounds on the *final* subgradient norm was easier than for the *smallest* subgradient norm among all iterates. Even without the *smallest* subgradient norm bounds, the bounds on the *final* subgradient norm in Table 5 (compared to Table 4) provide some insights (beyond (8)) on the relationship between the bounds on the subgradient norm and the composite gradient mapping norm as discussed in Sec. 5.4. We leave further modifying the code in [29] for computing tight bounds on the *smallest* subgradient norm or other criteria as future work.

PEP over both the cost function and the norm of the composite gradient mapping. Furthermore, we optimized the step coefficients of FSFOM with respect to the composite gradient mapping form of the (relaxed) PEP, yielding FPGM-OCG, which could be useful particularly when tackling dual problems.

Our relaxed PEP provided tractable analysis of the optimized step coefficients of FSFOM with respect to the cost function and the norm of the composite gradient mapping, but the relaxation is not guaranteed to be tight and the corresponding accelerations of PGM (FPGM and FPGM-OCG) are thus unlikely to be optimal. Therefore, finding optimal step coefficients of FSFOM over the cost function and the norm of the composite gradient mapping remain as future work. Nevertheless, the proposed FPGM-OCG that optimizes the composite gradient mapping form of the relaxed PEP and the FPGM-a (for any $a > 2$) may be useful in dual problems.

Software

Matlab codes for the SDP approaches in Sec. 3.4, Sec. 4.3 and Sec. 5 are available at https://gitlab.eecs.umich.edu/michigan-fast-optimization.

Acknowledgments

The authors would like to thank the anonymous referees for very useful comments that have improved the quality of this paper.

Funding: This research was supported in part by NIH grant U01 EB018753.

Appendix A

Derivation of the dual formulation (D) of (P1)

The derivation below is similar to [11, Lemma 2].

We replace $\max_{\boldsymbol{G},\boldsymbol{\delta}} LR^2\delta_{N-1}$ of (P1) by $\min_{\boldsymbol{G},\boldsymbol{\delta}}\{-\delta_{N-1}\}$ for convenience in this section. The corresponding dual function of such (P1) is then defined as

$$H(\boldsymbol{\lambda},\boldsymbol{\tau};\boldsymbol{h}) = \min_{\substack{\boldsymbol{G}\in\mathbb{R}^{N\times d},\\ \boldsymbol{\delta}\in\mathbb{R}^{N}}} \{\mathscr{L}(\boldsymbol{G},\boldsymbol{\delta},\boldsymbol{\lambda},\boldsymbol{\tau};\boldsymbol{h}) \coloneqq \mathscr{L}_1(\boldsymbol{\delta},\boldsymbol{\lambda},\boldsymbol{\tau}) + \mathscr{L}_2(\boldsymbol{G},\boldsymbol{\lambda},\boldsymbol{\tau};\boldsymbol{h})\}$$

for dual variables $\boldsymbol{\lambda} = [\lambda_1,\cdots,\lambda_{N-1}]^\top \in \mathbb{R}_+^{N-1}$ and $\boldsymbol{\tau} = [\tau_0,\cdots,\tau_{N-1}]^\top \in \mathbb{R}_+^{N}$, where $\mathscr{L}(\boldsymbol{G},\boldsymbol{\delta},\boldsymbol{\lambda},\boldsymbol{\tau};\boldsymbol{h})$ is a Lagrangian function, and

$$\mathscr{L}_1(\boldsymbol{\delta},\boldsymbol{\lambda},\boldsymbol{\tau}) \coloneqq -\delta_{N-1} + \sum_{i=1}^{N-1}\lambda_i(\delta_i - \delta_{i-1}) + \sum_{i=0}^{N1}\tau_i\delta_i,$$

$$\mathscr{L}_2(\boldsymbol{G}, \boldsymbol{\lambda}, \boldsymbol{\tau}; \boldsymbol{h}) \coloneqq \sum_{i=1}^{N-1} \lambda_i \operatorname{Tr}\{\boldsymbol{G}^\top \breve{\boldsymbol{A}}_{i-1,i}(\boldsymbol{h})\boldsymbol{G}\} + \sum_{i=0}^{N-1} \tau_i \operatorname{Tr}\{\boldsymbol{G}^\top \breve{\boldsymbol{D}}_i(\boldsymbol{h})\boldsymbol{G} + \boldsymbol{\nu}\boldsymbol{u}_i^\top \boldsymbol{G}\}.$$

Here, $\min_{\boldsymbol{\delta}} \mathscr{L}_1(\boldsymbol{\delta}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = 0$ for any $(\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda$ where Λ is defined in (23), and $\min_{\boldsymbol{\delta}} \mathscr{L}_1(\boldsymbol{\delta}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = -\infty$ otherwise.

For any given unit vector $\boldsymbol{\nu}$, [11, Lemma 1] implies

$$\min_{\boldsymbol{G} \in \mathbb{R}^{N \times d}} \mathscr{L}_2(\boldsymbol{G}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = \min_{\boldsymbol{w} \in \mathbb{R}^N} \mathscr{L}_2(\boldsymbol{w}\boldsymbol{\nu}^\top, \boldsymbol{\lambda}, \boldsymbol{\tau}),$$

and thus for any $(\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda$, we can rewrite the dual function as

$$H(\boldsymbol{\lambda}, \boldsymbol{\tau}; \boldsymbol{h}) = \min_{\boldsymbol{w} \in \mathbb{R}^N} \{\boldsymbol{w}^\top \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})\boldsymbol{w} + \boldsymbol{\tau}^\top \boldsymbol{w}\} = \max_{\gamma \in \mathbb{R}} \left\{ -\frac{1}{2}\gamma : \boldsymbol{w}^\top \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})\boldsymbol{w} + \boldsymbol{\tau}^\top \boldsymbol{w} \geq -\frac{1}{2}\gamma, \forall \boldsymbol{w} \in \mathbb{R}^N \right\} = \max_{\gamma \in \mathbb{R}} \left\{ -\frac{1}{2}\gamma : \begin{pmatrix} \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq 0 \right\},$$

where $\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})$ is defined in (24). Therefore the dual problem of (P1) becomes (D), recalling that we previously replaced $\max_{\boldsymbol{G}, \boldsymbol{\delta}} LR^2 \delta_{N-1}$ of (P1) by $\min_{\boldsymbol{G}, \boldsymbol{\delta}} \{-\delta_{N-1}\}$.

Appendix B

Proof of Prop. 5

The proof is similar to [17, Prop. 2, 3 and 4].

We first show that $\{h_{i+1,k}\}$ in (25) is equivalent to

$$h_{i+1,k} = \begin{cases} \dfrac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}} h_{i,k} & i = 0, \ldots, N-1, k = 0, \ldots, i-2, \\ \dfrac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}} (h_{i,i-1} - 1), & i = 0, \ldots, N-1, k = i-1, \\ 1 + \dfrac{(t_i - 1)t_{i+1}}{T_{i+1}}, & i = 0, \ldots, N-1, k = i, \end{cases} \tag{48}$$

We use the notation $h'_{i,k}$ for the coefficients (25) to distinguish from (48). It is obvious that $h'_{i+1,i} = h_{i+1,i}$, $i = 0, \ldots, N-1$, and we clearly have

$$h'_{i+1,i-1} = \frac{t_{i+1}}{T_{i+1}}(t_{i-1} - h'_{i,i-1}) = \frac{t_{i+1}}{T_{i+1}}\left(t_{i-1} - \left(1 + \frac{(t_{i-1}-1)t_i}{T_i}\right)\right) = \frac{(t_{i-1}-1)(T_i - t_i)t_{i+1}}{T_i T_{i+1}}$$
$$= \frac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}}(h_{i,i-1} - 1) = h_{i+1,i-1}$$

We next use induction by assuming $h'_{i+1,k} = h_{i+1,k}$ for $i = 0, \ldots, n-1$, $k = 0, \ldots, i$. We then have

$$h'_{n+1,k} = \frac{t_{n+1}}{T_{n+1}}\left(t_k - \sum_{j=k+1}^{n} h'_{j,k}\right) = \frac{t_{n+1}}{T_{n+1}}\left(t_k - \sum_{j=k+1}^{n-1} h'_{j,k} - h'_{n,k}\right) = \frac{t_{n+1}}{T_{n+1}}\left(\frac{T_n}{t_n}h'_{n,k} - h'_{n,k}\right)$$
$$= \frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}h_{n,k} = h_{n+1,k}$$

Next, using (48), we show that FSFOM with (25) is equivalent to the GFPGM. We use induction, and for clarity, we use the notation $\boldsymbol{y}'_0, \cdots, \boldsymbol{y}'_N$ for FSFOM with (48). It is obvious that $\boldsymbol{y}'_0 = \boldsymbol{y}_0$, and we have

$$\boldsymbol{y}'_1 = \boldsymbol{y}'_0 - \frac{1}{L}h_{1,0}\widetilde{\nabla}_L F(\boldsymbol{y}'_0) = \boldsymbol{y}_0 - \frac{1}{L}\left(1 + \frac{(t_0-1)t_1}{T_1}\right)\widetilde{\nabla}_L F(\boldsymbol{y}_0) = \boldsymbol{x}_1 + \frac{(T_0 - t_0)t_1}{t_0 T_1}(\boldsymbol{x}_1 - \boldsymbol{x}_0) + \frac{(t_0^2 - T_0)t_1}{t_0 T_1}(\boldsymbol{x}_1$$
$$- \boldsymbol{y}_0) = \boldsymbol{y}_1,$$

since $T_0 = t_0$. Assuming $\boldsymbol{y}'_i = \boldsymbol{y}_i$ for $i = 0, \ldots, n$, we then have

$$\boldsymbol{y}'_{n+1} = \boldsymbol{y}'_n - \frac{1}{L}h_{n+1,n}\widetilde{\nabla}_L F(\boldsymbol{y}'_n) - \frac{1}{L}h_{n+1,n-1}\widetilde{\nabla}_L F(\boldsymbol{y}'_{n-1}) - \frac{1}{L}\sum_{k=0}^{n-2} h_{n+1,k}\widetilde{\nabla}_L F(\boldsymbol{y}'_k) = \boldsymbol{y}_n - \frac{1}{L}\Bigg(1$$
$$+ \frac{(t_n - 1)t_{n+1}}{T_{n+1}}\Bigg)\widetilde{\nabla}_L F(\boldsymbol{y}_n) - \frac{1}{L}\frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}(h_{n,n-1} - 1)\widetilde{\nabla}_L F(\boldsymbol{y}_{n-1}) - \frac{1}{L}\sum_{k=0}^{n-2}\frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}h_{n,k}\widetilde{\nabla}_L F(\boldsymbol{y}_k$$
$$) = \boldsymbol{x}_{n+1} - \frac{1}{L}\frac{(t_n^2 - T_n)t_{n+1}}{t_n T_{n+1}}\widetilde{\nabla}_L F(\boldsymbol{y}_n) - \frac{1}{L}\frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}\left(\widetilde{\nabla}_L F(\boldsymbol{y}_n) - \widetilde{\nabla}_L F(\boldsymbol{y}_{n-1}) + \sum_{k=0}^{n-1} h_{n,k}\widetilde{\nabla}_L F(\boldsymbol{y}_k)\right)$$
$$= \boldsymbol{x}_{n+1} + \frac{(t_n^2 - T_n)t_{n+1}}{t_n T_{n+1}}(\boldsymbol{x}_{n+1} - \boldsymbol{y}_n) + \frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}\left(-\frac{1}{L}\widetilde{\nabla}_L F(\boldsymbol{y}_n) + \frac{1}{L}\widetilde{\nabla}_L F(\boldsymbol{y}_{n-1}) + \boldsymbol{y}_n - \boldsymbol{y}_{n-1}\right)$$
$$= \boldsymbol{x}_{n+1} + \frac{(T_n - t_n)t_{n+1}}{t_n T_{n+1}}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) + \frac{(t_n^2 - T_n)t_{n+1}}{t_n T_{n+1}}(\boldsymbol{x}_{n+1} - \boldsymbol{y}_n) = \boldsymbol{y}_{n+1}.$$

Appendix C

Proof of Prop. 8

The proof is similar to [17, Prop. 1 and 5].

We use induction, and for clarity, we use the notation $y'_0, \cdots, y'_N$ for FSFOM with (25) that is equivalent to GFPGM by Prop. 5. It is obvious that $y'_0 = y_0$, and we have

$$y'_1 = y'_0 - \frac{1}{L} h_{1,0} \widetilde{\nabla}_L F(y'_0) = y_0 - \frac{1}{L}\left(1 + \frac{(t_0 - 1)t_1}{T_1}\right)\widetilde{\nabla}_L F(y_0) = \left(1 - \frac{t_1}{T_1}\right)\left(y_0 - \frac{1}{L}\widetilde{\nabla}_L F(y_0)\right) + \frac{t_1}{T_1}\left(y_0 - \frac{1}{L} t_0 \widetilde{\nabla}_L F(y_0)\right) = \left(1 - \frac{t_1}{T_1}\right) x_1 + \frac{t_1}{T_1} z_1 = y_1 .$$

Assuming $y'_0 = y_i$ for $i = 0, \ldots, n$, we then have

$$y'_{n+1} = y'_n - \frac{1}{L} h_{n+1,n} \widetilde{\nabla}_L F(y'_n) - \frac{1}{L} \sum_{k=0}^{n-1} h_{n+1,k} \widetilde{\nabla}_L F(y'_k) = y_n - \frac{1}{L}\left(1 + \frac{(t_n - 1)t_{n+1}}{T_{n+1}}\right)\widetilde{\nabla}_L F(y_n) - \frac{1}{L}\sum_{k=0}^{n-1} \frac{t_{n+1}}{T_{n+1}}\left(t_k - \sum_{j=k+1}^{n} h_{j,k}\right)\widetilde{\nabla}_L F(y_k) = \left(1 - \frac{t_{n+1}}{T_{n+1}}\right)\left(y_n - \frac{1}{L}\widetilde{\nabla}_L F(y_n)\right) + \frac{t_{n+1}}{T_{n+1}}\left(y_n - \frac{1}{L}\sum_{k=0}^{n} t_k \widetilde{\nabla}_L F(y_k) + \frac{1}{L}\sum_{k=0}^{n-1}\sum_{j=k+1}^{n} h_{j,k}\widetilde{\nabla}_L F(y_k)\right) = \left(1 - \frac{t_{n+1}}{T_{n+1}}\right)\left(y_n - \frac{1}{L}\widetilde{\nabla}_L F(y_n)\right) + \frac{t_{n+1}}{T_{n+1}}\left(y_0 - \frac{1}{L}\sum_{k=0}^{n} t_k \widetilde{\nabla}_L F(y_k)\right) = \left(1 - \frac{t_{n+1}}{T_{n+1}}\right) x_{n+1} + \frac{t_{n+1}}{T_{n+1}} z_{n+1} .$$

Appendix D

Discussion on the choice of Ω_N in Sec. 4.1

Our formulation (P′) examines the set $\Omega_N = \{y_0, \cdots, y_{N-1}, x_N\}$ and eventually leads to the best known analytical bound on the norm of the composite gradient mapping in Thm. 11 among fixed-step first-order methods.

An alternative formulation would be to use the set $\{y_0, \cdots, y_{N-1}\}$ (i.e., excluding the point x_N). For this alternative, we could simply replace the inequality (33) with the condition $0 \leq F(y_{N-1}) - F(x_*)$ to derive a slightly different relaxation. (One could use other conditions at the point y_{N-1} as in [29] for a tight relaxation, but this is beyond the scope of this paper.) We found that the corresponding (loose) relaxation (P1′) using $\{y_0, \cdots, y_{N-1}\}$ leads to a larger upper bound than (40) in Thm. 10 for the set Ω_N.

Another alternative would be to use the set $\{x_0, \cdots, x_N\}$, which we leave as future work. Nevertheless, the inequality in Lemma 1 provides a bound for that set $\{x_0, \cdots, x_N\}$ as seen in Thm. 10 and 11.

We could also consider the final point x_N (or y_N) in (P′) instead of the minimum over a set of points. However, the corresponding (loose) relaxation (P1′) yielded only an $O(1/N)$ bound at best (even for the corresponding optimized step coefficients of (HD′)) on the final composite gradient mapping norm. So we leave finding its tighter relaxation as future work. Note that Table 4 reports tight numerical bounds on the composite gradient mapping norm at the final point x_N of algorithms considered.

Appendix E

Proof of Equation (45) in Thm. 11

$$\sum_{k=0}^{N-1}(T_k - t_k^2) + T_{N-1} = \sum_{k=m}^{N-1}\left(t_{m-1}^2 + \sum_{l=m}^{k} t_l - t_k^2\right) + t_{m-1}^2 + \sum_{l=m}^{N-1} t_l = (N-m+1)t_{m-1}^2 + \sum_{k=m}^{N-1}$$
$$\left(\sum_{l=m}^{k}\frac{N-l+1}{2} - \left(\frac{N-k+1}{2}\right)^2\right) + \sum_{l=m}^{N-1}\frac{N-l+1}{2} = (N-m+1)t_{m-1}^2 + \sum_{k'=0}^{N-m-1}\left(\sum_{l'=0}^{k'}\frac{N-l'-m+1}{2}\right.$$
$$\left. - \left(\frac{N-k'-m+1}{2}\right)^2\right) + \sum_{l'=0}^{N-m-1}\frac{N-l'-m+1}{2} = (N-m+1)t_{m-1}^2 + \sum_{k=0}^{N-m-1}\left(\frac{(N-m+1)(k+1)}{2}\right.$$
$$\left. - \frac{k(k+1)}{4} - \frac{(N-m+1)^2 - 2(N-m+1)k + k^2}{4}\right) + \frac{(N-m+1)(N-m)}{2} - \frac{(N-m-1)(N-m)}{4} = (N-m+1$$
$$)t_{m-1}^2 + \sum_{k=0}^{N-m-1}\left(-\frac{k^2}{2} + (N-m+3/4)k - \frac{(N-m-1)(N-m+1)}{4}\right) + \frac{(N-m)(N-m+3)}{4} = (N-m+1$$
$$)t_{m-1}^2 - \frac{(N-m-1)(N-m-1/2)(N-m)}{6} + \frac{(N-m-1)(N-m)(N-m+3/4)}{2}$$
$$- \frac{(N-m-1)(N-m)(N-m+1)}{4} + \frac{(N-m)(N-m+3)}{4} \geq \frac{(N-m+1)(m+1)^2}{4} + \frac{(N-m-1)(N-m)^2}{3}$$
$$- \frac{(N-m)^2(N-m+1)}{4} \geq \frac{(N-m-1)(N-m)^2}{3} \geq \frac{1}{24}(N-2)N^2,$$

where $m = \lfloor \frac{N}{2} \rfloor \geq \frac{N-1}{2}, N-m \geq \frac{N}{2}$, and $t_{m-1} \geq \frac{m+1}{2}$ (13).

References

1. Beck A, Nedic A, Ozdaglar A, Teboulle M. An O(1/k) gradient method for network resource allocation problems. IEEE Trans. Control of Network Systems. 2014; 1:64–73. DOI: 10.1109/TCNS.2014.2309751
2. Beck A, Teboulle M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Im. Proc. 2009; 18:2419–34. DOI: 10.1109/TIP.2009.2028250
3. Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. 2009; 2:183–202. DOI: 10.1137/080716542
4. Beck A, Teboulle M. A fast dual proximal gradient algorithm for convex minimization and applications. Operations Research Letters. 2014; 42:1–6. DOI: 10.1016/j.orl.2013.10.007
5. Chambolle A, Dossal C. On the convergence of the iterates of the “Fast iterative shrinkage/Thresholding algorithm”. J. Optim. Theory Appl. 2015; 166:968–82. DOI: 10.1007/s10957-015-0746-4
6. Combettes, PL., Pesquet, J-C. Proximal splitting methods in signal processing. Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, Optimization and Its Applications; 2011. p. 185-212.
7. CVX Research Inc. CVX: Matlab software for disciplined convex programming, version 2.0. Aug.. 2012 http://cvxr.com/cvx
8. Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Comm. Pure Appl. Math. 2004; 57:1413–57. DOI: 10.1002/cpa.20042
9. Devolder O, Glineur F, Nesterov Y. Double smoothing technique for large-scale linearly constrained convex optimization. SIAM J. Optim. 2012; 22:702–27. DOI: 10.1137/110826102

10. Drori Y. The exact information-based complexity of smooth convex minimization. Journal of Complexity. 2017; 39:1–16. DOI: 10.1016/j.jco.2016.11.001
11. Drori Y, Teboulle M. Performance of first-order methods for smooth convex minimization: A novel approach. Mathematical Programming. 2014; 145:451–82. DOI: 10.1007/s10107-013-0653-0
12. Drori Y, Teboulle M. An optimal variant of Kelley's cutting-plane method. Mathematical Programming. 2016; 160:321–51. DOI: 10.1007/s10107-016-0985-7
13. Ghadimi S, Lan G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Mathematical Programming. 2016; 156:59–99. DOI: 10.1007/s10107-015-0871-8
14. Goldstein T, O'Donoghue B, Setzer S, Baraniuk R. Fast alternating direction optimization methods. SIAM J. Imaging Sci. 2014; 7:1588–623. DOI: 10.1137/120896219
15. Grant, M., Boyd, S. Graph implementations for nonsmooth convex programs. In: Blondel, V.Boyd, S., Kimura, H., editors. Recent Advances in Learning and Control. Lecture Notes in Control and Information Sciences, Springer-Verlag Limited; 2008. p. 95-110.http://stanford.edu/~boyd/graph_dcp.html
16. Kim, D., Fessler, JA. Generalizing the optimized gradient method for smooth convex minimization. 2016. http://arxiv.org/abs/1607.06764. arxiv 1607.06764
17. Kim D, Fessler JA. Optimized first-order methods for smooth convex minimization. Mathematical Programming. 2016; 159:81–107. DOI: 10.1007/s10107-015-0949-3 [PubMed: 27765996]
18. Kim D, Fessler JA. On the convergence analysis of the optimized gradient method. J. Optim. Theory Appl. 2017; 172:187–205. DOI: 10.1007/s10957-016-1018-7 [PubMed: 28461707]
19. Lessard L, Recht B, Packard A. Analysis and design of optimization algorithms via integral quadratic constraints. SIAM J. Optim. 2016; 26:57–95. DOI: 10.1137/15M1009597
20. Löfberg, J. Yalmip : A toolbox for modeling and optimization in matlab. In Proceedings of the CACSD Conference; Taipei, Taiwan. 2004.
21. Monteiro RDC, Svaiter BF. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. SIAM J. Optim. 2013; 23:1092–1125. DOI: 10.1137/110833786
22. Necoara I, Patrascu A. Iteration complexity analysis of dual first order methods for conic convex programming. Optimization Methods and Software. 2016; 31:645–78. DOI: 10.1080/10556788.2016.1161763
23. Nesterov Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. Dokl. Akad. Nauk. USSR. 1983; 269:543–7.
24. Nesterov, Y. Introductory lectures on convex optimization: A basic course. Kluwer; 2004. http://books.google.com/books?id=VyYLem-l3CgC
25. Nesterov Y. Smooth minimization of non-smooth functions. Mathematical Programming. 2005; 103:127–52. DOI: 10.1007/s10107-004-0552-5
26. Nesterov Y. How to make the gradients small. Optima. 2012; 88:10–11.
27. Nesterov Y. Gradient methods for minimizing composite functions. Mathematical Programming. 2013; 140:125–61. DOI: 10.1007/s10107-012-0629-5
28. Sturm J. Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones. Optim. Meth. Software. 1999; 11:625–53. DOI: 10.1080/10556789908805766
29. Taylor AB, Hendrickx JM, Glineur F. Exact worst-case performance of first-order methods for composite convex optimization. SIAM J. Optim. 2017; 27:1283–1313. DOI: 10.1137/16M108104X
30. Taylor AB, Hendrickx JM, Glineur F. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. Mathematical Programming. 2017; 161:307–45. DOI: 10.1007/s10107-016-1009-3

Table 1

Asymptotic worst-case bounds on the cost function $F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*)$ and the norm of the composite gradient mapping $\min_{\boldsymbol{x}\in\Omega_N} \|\tilde{\nabla}_L F(\boldsymbol{x})\|$ of PGM, FPGM, FPGM-σ, FPGM-m, FPGM-OCG, and FPGM-a. (The cost function bound for FPGM-m in the table corresponds to the bound for FPGM after m iterations because a tight bound for the final Nth iteration is unknown. The bound on $\min_{i\in\{0,\ldots,N\}} \|\tilde{\nabla}_{L/\sigma^2} F(\boldsymbol{y}_i)\|$ is used for FPGM-σ.)

Algorithm	Asymptotic worst-case bound		Require selecting N in advance
	Cost function ($\times LR^2$)	Proximal gradient ($\times LR$)	
PGM	$\frac{1}{2}N^{-1}$	$2N^{-1}$	No
FPGM	$\mathbf{2N^{-2}}$	$2N^{-1}$	No
FPGM-σ ($0 < \sigma < 1$)	$\frac{2}{\sigma^2}N^{-2}$	$\frac{2\sqrt{3}}{\sigma^2}\sqrt{\frac{1+\sigma}{1-\sigma}}N^{-\frac{3}{2}}$	No
FPGM-($\sigma = 0.78$)	$3.3N^{-2}$	$16.2N^{-\frac{3}{2}}$	
FPGM$-\left(m = \lfloor\frac{2N}{3}\rfloor\right)$	$4.5N^{-2}$	$5.2N^{-\frac{3}{2}}$	Yes
FPGM-OCG	$4N^{-2}$	$\mathbf{4.9N^{-\frac{3}{2}}}$	Yes
FPGM-a ($a > 2$)	aN^{-2}	$\frac{a\sqrt{6}}{\sqrt{a-2}}N^{-\frac{3}{2}}$	No
FPGM-($a = 4$)	$4N^{-2}$	$6.9N^{-\frac{3}{2}}$	

Table 2

Tight worst-case bounds on the cost function $LR^2/(F(\boldsymbol{x}_N) - F(\boldsymbol{x}_*))$ of PGM, FPGM, FPGM-($\sigma = 0.78$), FPGM–$\left(m = \lfloor \frac{2N}{3} \rfloor\right)$, FPGM-OCG, and FPGM-($a = 4$). We computed empirical rates by assuming that the bounds follow the form bN^c with constants b and c, and then by estimating c from points $N = 47$, 50. Note that the corresponding empirical rates are underestimated due to the simplified exponential model.

N	PGM	FPGM	FPGM -($\sigma = 0.78$)	FPGM–$\left(m = \lfloor \frac{2N}{3} \rfloor\right)$	FPGM -OCG	FPGM -($a = 4$)
1	4.00	4.00	2.43	4.00	4.00	4.00
2	8.00	8.00	4.87	8.00	8.00	8.00
4	16.00	19.35	11.77	17.13	17.60	17.23
10	40.00	79.07	48.11	56.47	59.25	55.88
20	80.00	261.66	159.19	163.75	170.10	159.17
30	120.00	546.51	332.49	321.56	331.97	312.03
40	160.00	932.89	567.57	502.37	544.55	514.73
47	188.00	1263.58	768.76	675.68	723.06	686.33
50	200.00	1420.45	864.20	752.90	807.66	767.37
Empi. $O(\cdot)$	$N^{-1.00}$	$N^{-1.89}$	$N^{-1.89}$	$N^{-1.75}$	$N^{-1.79}$	$N^{-1.80}$
Known $O(\cdot)$	N^{-1}	N^{-2}	N^{-2}	N^{-2}	N^{-2}	N^{-2}

Table 3

Tight worst-case bounds on the norm of the composite gradient mapping $LR/(\min_{\boldsymbol{x}\in\Omega_N}\|\tilde{\nabla}_L F(\boldsymbol{x})\|)$ of PGM, FPGM, FPGM-($\sigma = 0.78$), FPGM–$\left(m = \lfloor\frac{2N}{3}\rfloor\right)$, FPGM-OCG, and FPGM-($a = 4$). Empirical rates were computed as described in Table 2. (The bound for FPGM-σ uses $\min_{\boldsymbol{x}\in\Omega_N}\|\tilde{\nabla}_{L/\sigma^2} F(\boldsymbol{x})\|$.)

N	**PGM**	**FPGM**	**FPGM -($\sigma = 0.78$)**	FPGM–$\left(m = \lfloor\frac{2N}{3}\rfloor\right)$	**FPGM -OCG**	**FPGM -(a=4)**
1	1.84	1.84	1.18	1.84	1.84	1.84
2	2.83	2.83	1.78	2.83	2.83	2.83
4	4.81	5.65	3.50	5.09	5.21	5.12
10	10.80	13.24	8.74	14.91	15.60	14.76
20	20.78	27.19	18.83	39.70	39.61	29.21
30	30.78	43.49	30.82	64.45	64.40	47.14
40	40.78	61.76	44.39	92.82	91.99	67.82
47	47.77	75.60	54.73	113.92	113.41	83.67
50	50.77	81.78	59.35	123.54	123.17	90.78
Empi. $O(\cdot)$	$N^{-0.98}$	$N^{-1.27}$	$N^{-1.31}$	$N^{-1.31}$	$N^{-1.33}$	$N^{-1.32}$
Known $O(\cdot)$	N^{-1}	N^{-1}	$N^{-\frac{3}{2}}$	$N^{-\frac{3}{2}}$	$N^{-\frac{3}{2}}$	$N^{-\frac{3}{2}}$

SIAM J Optim. Author manuscript; available in PMC 2018 May 23.

Table 4

Tight worst-case bounds on the norm of the final composite gradient mapping $LR/\|\tilde{\nabla}_L F(\boldsymbol{x}_N)\|$ of PGM, FPGM, FPGM-($\sigma = 0.78$), FPGM$-\left(m = \lfloor \frac{2N}{3} \rfloor\right)$, FPGM-OCG, and FPGM-($a = 4$). Empirical rates were computed as described in Table 2. (The bound for FPGM-σ uses $\|\tilde{\nabla}_{L/\sigma^2} F(\boldsymbol{x}_N)\|$.)

N	PGM	FPGM	FPGM -($\sigma = 0.78$)	FPGM$-\left(m = \lfloor \frac{2N}{3} \rfloor\right)$	FPGM -OCG	FPGM -($a = 4$)
1	1.84	1.84	1.18	1.84	1.84	1.84
2	2.83	2.83	1.78	2.83	2.83	2.83
4	4.81	5.65	3.50	5.09	5.21	5.12
10	10.80	12.68	8.41	14.91	15.60	14.76
20	20.78	22.02	14.26	39.65	39.10	25.96
30	30.78	31.26	20.12	64.40	63.40	34.21
40	40.78	40.46	25.97	92.78	90.16	42.39
47	47.77	46.89	30.06	113.92	110.12	48.13
50	50.77	49.65	31.81	123.53	118.99	50.59
Empi. $O(\cdot)$	$N^{-0.98}$	$N^{-0.92}$	$N^{-0.92}$	$N^{-1.31}$	$N^{-1.25}$	$N^{-0.81}$
Known $O(\cdot)$	N^{-1}	N^{-1}	N^{-1}	$N^{-\frac{3}{2}}$	N^{-1}	N^{-1}

SIAM J Optim. Author manuscript; available in PMC 2018 May 23.

Table 5

Tight worst-case bounds on the subgradient norm $LR/\|F'(x_N)\|$ of PGM, FPGM, FPGM-($\sigma = 0.78$), FPGM–$\left(m = \lfloor \frac{2N}{3} \rfloor\right)$, FPGM-OCG, and FPGM-($a = 4$), where $F'(x) \in \partial F(x)$ is a subgradient. Empirical rates were computed as described in Table 2.

N	PGM	FPGM	FPGM -($\sigma = 0.78$)	FPGM–$\left(m = \lfloor \frac{2N}{3} \rfloor\right)$	FPGM -OCG	FPGM -($a = 4$)
1	1.00	1.00	0.61	1.00	1.00	1.00
2	2.00	2.00	1.22	2.00	2.00	2.00
4	4.00	4.83	2.94	4.28	4.40	4.31
10	10.00	7.60	4.67	14.12	14.81	12.10
20	20.00	12.58	7.67	38.29	36.65	16.85
30	30.00	17.63	10.74	62.71	60.40	21.61
40	40.00	22.67	13.80	91.00	86.62	26.47
47	47.00	26.20	15.94	112.01	106.21	29.91
50	50.00	27.71	16.86	121.53	114.93	31.39
Empi. $O(\cdot)$	$N^{-1.00}$	$N^{-0.91}$	$N^{-0.91}$	$N^{-1.32}$	$N^{-1.27}$	$N^{-0.78}$
Known $O(\cdot)$	N^{-1}	N^{-1}	N^{-1}	$N^{-\frac{3}{2}}$	N^{-1}	N^{-1}

SIAM J Optim. Author manuscript; available in PMC 2018 May 23.