SIAM J. OPTIM.
Vol. 28, No. 2, pp. 1920–1950

© 2018 Society for Industrial and Applied Mathematics

GENERALIZING THE OPTIMIZED GRADIENT METHOD FOR SMOOTH CONVEX MINIMIZATION[*]

DONGHWAN KIM[†] AND JEFFREY A. FESSLER[†]

Abstract. This paper generalizes the optimized gradient method (OGM) [Y. Drori and M. Teboulle, *Math. Program.*, 145 (2014), pp. 451–482], [D. Kim and J. A. Fessler, *Math. Program.*, 159 (2016), pp. 81–107], [D. Kim and J. A. Fessler, *J. Optim. Theory Appl.*, 172 (2017), pp. 187–205] that achieves the optimal worst-case cost function bound of first-order methods for smooth convex minimization [Y. Drori, *J. Complexity*, 39 (2017), pp. 1–16]. Specifically, this paper studies a generalized formulation of OGM and analyzes its worst-case rates in terms of both the function value and the norm of the function gradient. This paper also develops a new algorithm called OGM-OG that is in the generalized family of OGM and that has the best known analytical worst-case bound with rate $O(1/N^{1.5})$ on the decrease of the gradient norm among fixed-step first-order methods. This paper also proves that Nesterov's fast gradient method [Y. Nesterov, *Dokl. Akad. Nauk. USSR*, 269 (1983), pp. 543–547], [Y. Nesterov, *Math. Program.*, 103 (2005), pp. 127–152] has an $O(1/N^{1.5})$ worst-case gradient norm rate but with constant larger than OGM-OG. The proof is based on the worst-case analysis called the Performance Estimation Problem in [Y. Drori and M. Teboulle, *Math. Program.*, 145 (2014), pp. 451–482].

Key words. first-order algorithms, gradient methods, smooth convex minimization, worst-case performance analysis

AMS subject classifications. 90C25, 90C30, 90C60, 68Q25, 49M25, 90C22

DOI. 10.1137/17M112124X

1. Introduction.

First-order methods are favorable for solving large-scale problems because their computational complexity per iteration depends mildly on the problem dimension. In particular, Nesterov's fast gradient method (FGM) [24, 26] achieves the optimal worst-case rate $O(1/N^2)$ for decreasing smooth convex functions after N iterations [25], and thus has been widely used in (large-scale) applications. Recently, the optimized gradient method (OGM) [9, 15, 16] was found to achieve the optimal worst-case cost function bound of first-order methods (with either fixed-step or adaptive-step approaches) for smooth convex minimization in [7], whereas FGM achieves that bound only up to a constant.[1]

Building upon [9, 15, 16], this paper presents two different ways of generalizing OGM and its development. First, this paper specifies a parameterized family of algorithms that generalizes OGM and provides worst-case bounds on the function and gradient norm values for this family. Like the generalized forms of FGM [3, 26] being widely used and studied (e.g., [1, 3, 29]), we believe introducing the generalized OGM here can be potentially useful. Second, this paper optimizes the step coefficients of fixed-step first-order methods with respect to the rate of decrease of the cost func-

[*] Received by the March 16, 2017; accepted for publication (in revised form) April 24, 2018; published electronically June 28, 2018.

http://www.siam.org/journals/siopt/28-2/M112124.html

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

[†] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 (kimdongh@umich.edu, fessler@umich.edu).

[1] There is a backtracking line-search version of FGM [25] that also achieves the optimal worst-case function bound up to constant, which is sometimes more useful than the fixed-step FGM in practice. However, such a backtracking line-search version of OGM with a fast worst-case bound is yet unknown (unlike the fixed-step OGM [9, 15, 16]), while recently an exact line-search version of OGM is developed in [8].

tion's gradient norm, leading to a new algorithm called OGM-OG (OG for "optimized over a gradient"). This development expands the choice of worst-case rate metrics for optimizing first-order methods in [9, 15, 16] that focused on the cost function decrease leading to OGM. We next briefly review the Performance Estimation Problem (PEP) [9] that was used in [9, 15, 16] to develop OGM and that we extensively use throughout the paper.

Drori and Teboulle [9] cast a worst-case analysis into an optimization problem called PEP[2] [9] that examines the maximal absolute cost function inaccuracy over all possible inputs (cost functions) to the optimization algorithm. (See, e.g., [5, 8, 10, 15, 16, 17, 18, 19, 30, 31] for its extensions.) Moreover, Drori and Teboulle [9] optimized *numerically* the step coefficients of first-order methods using PEP for smooth convex minimization and found an algorithm whose worst-case bound is lower than that of FGM, but which required too much computation and memory to be appealing for large-scale problems. Building on their work, the authors [15, 16] found a computationally and memorywise efficient version, called OGM, and showed analytically that OGM satisfies an analytical worst-case bound that is twice smaller than that of FGM. Drori [7] showed that the OGM is optimal for large-dimensional smooth convex minimization over a general class of first-order methods with either fixed or adaptive step sizes [7]. This OGM has been numerically extended for nonsmooth composite convex problems in [30]. In addition, this OGM-type algorithm was already studied in the context of a proximal point method [13, Appendix].

Using the PEP approach [9], this paper proposes a generalized version of OGM (GOGM) and analyzes its worst-case rates in terms of *both* the decrease of the cost function and the decrease of the norm of the gradient of the cost function. The results complement the worst-case analysis of the OGM [15, 16] and expand our understanding of OGM-type first-order methods.

This paper analyzes the worst-case rate of the gradient norm (in addition to that of the cost function) because it is important when dealing with dual problems, considering that the dual gradient norm corresponds to the primal distance to feasibility (see, e.g., [6, 22, 27]). While FGM has not been shown previously to satisfy a rate $O(1/N^{1.5})$ for decreasing the gradient norm, modified versions of FGM with such a rate were studied in [11, 21, 27]. This paper proves that FGM in fact does have that rate, building upon [31], which numerically conjectured such a rate for FGM using the gradient norm version of PEP. For further acceleration of the worst-case gradient norm rate, we optimize the step coefficients of first-order methods with respect to the gradient norm using PEP and propose an algorithm named OGM-OG that belongs to the GOGM family and has the best known analytical worst-case bound on rate of decrease of the gradient norm among fixed-step first-order methods.

One can extend some aspects of the approaches for generalizing OGM described in this paper to other optimization algorithms and problems. One direction we have already taken in [17] aims to improve the fast iterative shrinkage/thresholding algorithm (FISTA) [2] (that reduces to FGM for smooth convex problems) for nonsmooth composite convex problems. Naturally, this paper and [17] use some similar approaches, but they are different; the methods in [17] when simplified to the smooth case correspond to a generalization of FGM that differs from the GOGM. Another direction we have recently taken in [18] focuses on optimizing the step coefficients of first-order methods with respect to the gradient norm under the initial bounded

[2] The original PEP was intractable to solve, so a series of relaxations on the PEP was introduced in [9] to make it possibly solvable, which we review in section 4.1.

function condition that is different from the initial bounded distance condition used in this paper.

Section 2 defines the smooth convex problem and the first-order methods. Section 3 reviews and discusses worst-case analyses of a gradient method (GM), FGM, and OGM for both the function value and the gradient norm. Section 3 also reviews first-order methods that guarantee an $O(1/N^{1.5})$ rate for the gradient decrease. Section 4 reviews the cost function form of PEP [9] and reviews how the OGM [15] is derived using such PEP. Section 4 then proposes a generalized version of OGM (GOGM) using the cost function form of PEP, and section 5 provides a worst-case gradient norm bound for the GOGM using the gradient form of PEP. Then section 5 optimizes the step coefficients using the gradient form of PEP and proposes the OGM-OG that belongs to the GOGM family. Section 5 also proves that FGM decreases the gradient norm with a rate $O(1/N^{1.5})$. Sections 6 and 7 provide discussion and conclusion.

2. Smooth convex problem and first-order methods.

2.1. Smooth convex problem. We focus on the smooth convex minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}), \tag{M}$$

where the following additional conditions are assumed:

- $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function of the type $\mathcal{C}_L^{1,1}(\mathbb{R}^d)$, i.e., continuously differentiable with Lipschitz continuous gradient:
  $$||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})|| \le L||\boldsymbol{x} - \boldsymbol{y}|| \quad \text{for all} \;\; \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \tag{2.1}$$
  where $L > 0$ is the Lipschitz constant.
- 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 that satisfy the above conditions. We also assume that the distance between an initial point $\boldsymbol{x}_0$ and an optimal solution $\boldsymbol{x}_* \in X_*(f)$ is bounded by some $R > 0$, i.e.,

$$||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \le R. \tag{2.2}$$

2.2. First-order methods. To solve (M), we consider the following class of *fixed-step* (or nonadaptive-step) first-order methods (FSFOM), where the update step at the $(i+1)$th iteration is a weighted sum of the previous and current gradients $\{\nabla f(\boldsymbol{x}_k)\}_{k=0}^{i}$ scaled by $\frac{1}{L}$ with fixed constant step coefficients $\{h_{i+1,k}\}_{k=0}^{i}$ that are not adaptive to the given f and $\boldsymbol{x}_0$ (and thus L and R). This class FSFOM includes GM, FGM, OGM, and the methods proposed in this paper, but excludes line-search-type methods.

```
Algorithm Class FSFOM
  Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$.
  For $i = 0, \ldots, N-1$
      $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k}\nabla f(\boldsymbol{x}_k).$
```

3. Review of the worst-case analysis of FSFOM.

This section reviews the worst-case analysis of existing FSFOMs (and simple variants thereof) in terms of bounds on the cost function and gradient norm. Section 3.1 reviews the worst-case cost function decrease of GM, FGM, and OGM. Section 3.2 presents both FSFOMs (including GM, FGM, OGM, and some variants) that have either $O(1/N)$ or $O(1/N^{1.5})$ rate for the worst-case gradient decrease; it also reviews an $O(1/N^2)$ lower bound of the worst-case rates of first-order methods for decreasing the gradient norm.

3.1. Function value worst-case analysis of FSFOM.

The simplest example of an FSFOM is the following GM that uses only the current gradient and the Lipschitz constant L for the update.

```
Algorithm GM
  Input: $f \in \mathcal{F}_L(\mathbb{R}^d),\ \boldsymbol{x}_0 \in \mathbb{R}^d$.
  For $i = 0, 1, \ldots$
      $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
```

This GM monotonically decreases the cost function [25] and satisfies the following tight[3] worst-case bound [9, Thm. 1] for any $i \geq 0$:

$$f(\boldsymbol{x}_i) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{4i+2}. \tag{3.1}$$

Among the class FSFOM, the following two equivalent forms of FGM [24, 26] have been used widely because they decrease the cost function with the optimal rate $O(1/N^2)$.

```
Algorithm FGM1
Input: $f \in \mathcal{F}_L(\mathbb{R}^d),\ \boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $t_0 = 1$.
For $i = 0, 1, \ldots$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$
    $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1} + \frac{t_i - 1}{t_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i)$
```

```
Algorithm FGM2
Input: $f \in \mathcal{F}_L(\mathbb{R}^d),\ \boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $t_0 = 1$.
For $i = 0, 1, \ldots$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    $\boldsymbol{z}_{i+1} = \boldsymbol{x}_0 - \frac{1}{L}\sum_{k=0}^{i} t_k \nabla f(\boldsymbol{x}_k)$
    $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$
    $\boldsymbol{x}_{i+1} = \left(1 - \frac{1}{t_{i+1}}\right)\boldsymbol{y}_{i+1} + \frac{1}{t_{i+1}}\boldsymbol{z}_{i+1}$
```

Specifically, the FGM1 and FGM2 iterates satisfy the following worst-case cost function bounds [15, 24, 26] for any $i \geq 1$:

$$f(\boldsymbol{y}_i) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{2t_{i-1}^2} \leq \frac{2LR^2}{(i+1)^2} \quad \text{and} \quad f(\boldsymbol{x}_i) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{2t_i^2} \leq \frac{2LR^2}{(i+2)^2}, \tag{3.2}$$

[3] A *tight* worst-case bound denotes an inequality where the equality holds for some function f. For example, [9, Thm. 2] shows that the bound (3.1) is tight.

where the parameter t_i satisfies

$$t_i^2 = \sum_{l=0}^{i} t_l \quad \text{and} \quad t_i \geq \frac{i+2}{2} \quad \text{for all } i. \tag{3.3}$$

A generalized form of FGM in [3] uses parameters t_i satisfying $t_0 = 1$ and $t_i^2 \leq t_{i-1}^2 + t_i$, including the choice $t_i = \frac{i+a}{a}$ for any $a \geq 2$. There is another generalized form of FGM in [26], and these generalized forms of FGM have been widely used and studied (e.g., [1, 3, 29]). Similarly this paper studies generalizations of the OGM.

Building upon [9], which optimized numerically the step coefficients over the cost function form of PEP, the authors [15] developed the following two equivalent forms of OGM, as reviewed in section 4.

```
Algorithm OGM1
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$,
    $\theta_0 = 1$.
For $i = 0, \dots, N-1$
    $y_{i+1} = x_i - \frac{1}{L}\nabla f(x_i)$
    $\theta_{i+1} = \begin{cases} \frac{1+\sqrt{1+4\theta_i^2}}{2}, & i \leq N-2 \\ \frac{1+\sqrt{1+8\theta_i^2}}{2}, & i = N-1 \end{cases}$
    $x_{i+1} = y_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(y_{i+1} - y_i)$
                $+ \frac{\theta_i}{\theta_{i+1}}(y_{i+1} - x_i)$
```

```
Algorithm OGM2
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$,
    $\theta_0 = 1$.
For $i = 0, \dots, N-1$
    $y_{i+1} = x_i - \frac{1}{L}\nabla f(x_i)$
    $z_{i+1} = x_0 - \frac{1}{L}\sum_{k=0}^{i} 2\theta_k \nabla f(x_k)$
    $\theta_{i+1} = \begin{cases} \frac{1+\sqrt{1+4\theta_i^2}}{2}, & i \leq N-2 \\ \frac{1+\sqrt{1+8\theta_i^2}}{2}, & i = N-1 \end{cases}$
    $x_{i+1} = \left(1 - \frac{1}{\theta_{i+1}}\right) y_{i+1} + \frac{1}{\theta_{i+1}} z_{i+1}$
```

The OGM iterates satisfy the following worst-case cost function bounds [15, 16]:

$$f(\boldsymbol{y}_i) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{4\theta_{i-1}^2} \leq \frac{LR^2}{(i+1)^2} \tag{3.4}$$

for any $1 \leq i \leq N$, and

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{2\theta_N^2} \leq \frac{LR^2}{(N+1)(N+1+\sqrt{2})}. \tag{3.5}$$

The parameter sequence θ_i satisfies

$$\theta_i^2 = \begin{cases} \sum_{l=0}^{i+1} \theta_l, & i \leq N-1, \\ 2\sum_{l=0}^{N-1} \theta_l + \theta_N, & i = N, \end{cases} \quad \text{and} \quad \theta_i \geq \begin{cases} \frac{i+2}{2}, & i \leq N-1, \\ \frac{i+1}{\sqrt{2}}, & i = N, \end{cases} \tag{3.6}$$

which is equivalent to t_i (3.3) except at the final iteration. The bounds (3.4) and (3.5) of OGM are about twice as small as the bounds (3.2) of FGM, so OGM decreases the cost function faster than FGM in the worst case (and often in practice [14]). In addition, the bound (3.5) on the final iterate $\boldsymbol{x}_N$ is tight and satisfies the optimal worst-case bound of general first-order methods including both FSFOM and adaptive-step first-order methods, when the condition $d \geq N+1$ holds [7].

The additional term $\frac{\theta_i}{\theta_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i)$ of OGM1 and the additional constant 2 for the update of $\boldsymbol{z}_i$ of OGM2, compared to FGM1 and FGM2, respectively, along with the parameter θ_N, are what make OGM optimal (for $d \geq N+1$). One of the main goals of this paper is to generalize the form of OGM and analyze the worst-case rate of such generalized OGM in terms of both the function value and the gradient norm, complementing the bounds (3.4) and (3.5) on the function value of OGM.

The next section studies the worst-case rate of the gradient of FSFOM.

3.2. Gradient norm worst-case analysis of FSFOM. When tackling dual problems, it is known that the gradient norm worst-case rate is important in addition to the function value worst-case rate because the dual gradient norm is related to the primal distance to feasibility (see, e.g., [6, 22, 27]). One simple way to find a (loose) worst-case bound for the gradient norm is to use the well-known convex inequality for convex functions with L-Lipschitz continuous gradients [25]:

$$\frac{1}{2L}||\nabla f(\boldsymbol{x})||^2 \leq f(\boldsymbol{x}) - f\left(\boldsymbol{x} - \frac{1}{L}\nabla f(\boldsymbol{x})\right) \leq f(\boldsymbol{x}) - f(\boldsymbol{x}_*) \quad \text{for all } \ \boldsymbol{x} \in \mathbb{R}^d, \tag{3.7}$$

as discussed in [25, 31]. Combining the bounds (3.1), (3.2) and the inequality (3.7), for any $i \geq 1$, the GM iterates satisfy

$$||\nabla f(\boldsymbol{x}_i)|| \leq \sqrt{2L(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_*))} \leq \frac{LR}{\sqrt{2i+1}}, \tag{3.8}$$

and the iterates of FGM satisfy

$$||\nabla f(\boldsymbol{y}_i)|| \leq \frac{LR}{t_{i-1}} \leq \frac{2LR}{i+1} \quad \text{and} \quad ||\nabla f(\boldsymbol{x}_i)|| \leq \frac{LR}{t_i} \leq \frac{2LR}{i+2}. \tag{3.9}$$

Similarly for any $1 \leq i \leq N$, the OGM iterates with the bounds (3.4) and (3.5) satisfy

$$||\nabla f(\boldsymbol{y}_i)|| \leq \frac{LR}{\sqrt{2}\theta_{i-1}} \leq \frac{\sqrt{2}LR}{i+1} \quad \text{and} \quad ||\nabla f(\boldsymbol{x}_N)|| \leq \frac{LR}{\theta_N} \leq \frac{\sqrt{2}LR}{N+1}. \tag{3.10}$$

Unfortunately, using the inequality (3.7) provides at best an $O(1/N)$ bound due to the optimal rate $O(1/N^2)$ of the function decrease. Furthermore, in general using (3.7) need not lead to tight worst-case bounds on the gradient norm.

Using a different approach, a smaller $O(1/N)$ worst-case bound for the gradient norm of GM was derived in [27], as reviewed in the next section. While the bounds (3.8), (3.9), and (3.10) are not guaranteed to be tight, the next section shows that the worst-case gradient bound (3.10) on the final iterate $\boldsymbol{x}_N$ of OGM is in fact tight and thus has the same disappointingly slow $O(1/N)$ worst-case bound on the gradient norm as GM.

3.2.1. FSFOM with rate $O(1/N)$ for decreasing the gradient norm. This section uses the following lemma stating that GM monotonically decreases the gradient.

LEMMA 3.1 ([22, Lem. 2.4]). *The GM monotonically decreases the gradient norm, i.e.,*

$$\left|\left|\nabla f\left(\boldsymbol{x} - \frac{1}{L}\nabla f(\boldsymbol{x})\right)\right|\right| \leq ||\nabla f(\boldsymbol{x})||. \tag{3.11}$$

The following theorem reviews a simple proof in [27] that provides a worst-case gradient norm bound for GM with rate $O(1/N)$ that is smaller than (3.8), where [22, Thm. 6.1] additionally considers Lemma 3.1.

Theorem 3.2 ([22, Thm. 6.1], [27]). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by GM. Then, for any $N \geq 1$,*

$$\min_{i \in \{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)|| = ||\nabla f(\boldsymbol{x}_N)|| \leq \frac{\sqrt{2}LR}{\sqrt{N(N+2)}}. \tag{3.12}$$

Proof. Lemma 3.1 implies the first equality in (3.12). Using (3.1), (3.7), (3.11) yields

$$\begin{aligned} \frac{LR^2}{4m+2} &\overset{(3.1)}{\geq} f(\boldsymbol{x}_m) - f(\boldsymbol{x}_*) \overset{(3.7)}{\geq} f(\boldsymbol{x}_{N+1}) - f(\boldsymbol{x}_*) + \frac{1}{2L} \sum_{i=m}^{N} ||\nabla f(\boldsymbol{x}_i)||^2 \\ &\overset{(3.11)}{\geq} \frac{N-m+1}{2L} ||\nabla f(\boldsymbol{x}_N)||^2, \end{aligned}$$

which is equivalent to (3.12) using $m = \lfloor N/2 \rfloor$ for which $m \geq \frac{N-1}{2}$ and $N - m \geq \frac{N}{2}$. □

Inspired by the conjecture in [31, sect. 4.1.3], the following theorem shows that the $O(1/N)$ rate of the worst-case gradient norm bound (3.12) of GM is tight up to a constant.

Theorem 3.3. *Let $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by GM. Then, for any $N \geq 1$,*

$$\frac{LR}{N+1} \leq \max_{\substack{f \in \mathcal{F}_L(\mathbb{R}^d), \\ \boldsymbol{x}_* \in X_*(f), \\ ||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \leq R}} \min_{i \in \{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)|| = \max_{\substack{f \in \mathcal{F}_L(\mathbb{R}^d), \\ \boldsymbol{x}_* \in X_*(f), \\ ||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \leq R}} ||\nabla f(\boldsymbol{x}_N)||, \tag{3.13}$$

where the inequality in (3.13) *is achieved by the following function in $\mathcal{F}_L(\mathbb{R}^d)$:*

$$\psi(\boldsymbol{x}) = \begin{cases} \frac{LR}{N+1} ||\boldsymbol{x}|| - \frac{LR^2}{2(N+1)^2}, & ||\boldsymbol{x}|| \geq \frac{R}{N+1}, \\ \frac{L}{2} ||\boldsymbol{x}||^2, & ||\boldsymbol{x}|| < \frac{R}{N+1}. \end{cases} \tag{3.14}$$

Proof. Lemma 3.1 implies the equality in (3.13). Starting from $\boldsymbol{x}_0 = R\boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a unit vector, the GM iterates are

$$\boldsymbol{x}_i = \left(1 - \frac{i}{N+1}\right) R\boldsymbol{\nu}, \quad \nabla\psi(\boldsymbol{x}_i) = \frac{LR}{N+1}\boldsymbol{\nu}, \quad i = 0, \ldots, N,$$

which implies the inequality (3.13). □

We next show that the bound (3.10) for the gradient norm at the final iterate $\boldsymbol{x}_N$ of OGM is tight and that its worst-case function is a simple quadratic function. Note that OGM was derived by optimizing a worst-case bound on the cost function decrease, and its behavior in terms of gradient norms was not investigated previously.

Theorem 3.4. *Let $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by OGM. Then, for any $N \geq 1$,*

$$\max_{\substack{f \in \mathcal{F}_L(\mathbb{R}^d), \\ \boldsymbol{x}_* \in X_*(f), \\ ||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \leq R}} \min_{i \in \{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)|| = \max_{\substack{f \in \mathcal{F}_L(\mathbb{R}^d), \\ \boldsymbol{x}_* \in X_*(f), \\ ||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \leq R}} ||\nabla f(\boldsymbol{x}_N)|| = \frac{LR}{\theta_N} \left(\leq \frac{\sqrt{2}LR}{N+1}\right), \tag{3.15}$$

where the worst-case function in $\mathcal{F}_L(\mathbb{R}^d)$ for OGM in terms of the gradient norm is the quadratic function $\phi(\boldsymbol{x}) = \frac{L}{2}||\boldsymbol{x}||^2$.

Proof. See Appendix A. ∎

Comparing (3.12) and (3.15), we see that GM and OGM have essentially similar worst-case gradient norm bounds. This is a dilemma because OGM is the fastest FSFOM in terms of the worst-case cost function bound, but is as slow as GM in terms of the worst-case gradient norm bound. Therefore, one of the main goals of this paper is to study optimizing the step coefficients of FSFOM using PEP with respect to the gradient norm in section 5.4.

We next discuss the specific FSFOM in [27] that decreases the gradient norm with a faster $O(1/N^{1.5})$ rate.

3.2.2. FSFOM with rate $O(1/N^{1.5})$ for decreasing the gradient norm. Searching for an FSFOM that decreases the gradient norm faster than the $O(1/N)$ rate of GM (and OGM), Nesterov [27] (among other variants of FGM [11, 21]) considered performing FGM for the first m iterations, and GM for the remaining iterations. He showed that this method, which we denote FGM-m, satisfies a fast rate $O(1/N^{1.5})$ for decreasing the gradient norm. In [6, 22, 27], FGM-m for $m = \lfloor N/2 \rfloor$ was used to solve dual problems. To pursue a faster worst-case rate (in terms of the constant factor), we consider here another variant that performs OGM for the first m iterations and GM for the remaining iterations, which we denote OGM-m.

```
Algorithm OGM-m
  Input: $f \in \mathcal{F}_L(\mathbb{R}^d),\ \boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d,\ \vartheta_0 = 1,\ m \in \{1, \dots, N-1\}$.
  For $i = 0, \dots, m-1$
      $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
      $\vartheta_{i+1} = \begin{cases} \frac{1+\sqrt{1+4\vartheta_i^2}}{2}, & i \le m-2 \\ \frac{1+\sqrt{1+8\vartheta_i^2}}{2}, & i = m-1 \end{cases}$
      $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1} + \frac{\vartheta_i - 1}{\vartheta_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i) + \frac{\vartheta_i}{\vartheta_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i)$
  For $i = m, \dots, N-1$
      $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
```

The following theorem bounds the gradient norm of the OGM-m iterates, inspired by the proof in [22, 27] for the worst-case gradient norm bound of the FGM-m iterates. The worst-case bound of FGM-m in [22, 27] is asymptotically $\sqrt{2}$-times larger than the following new bound (3.16) for OGM-m.

THEOREM 3.5. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{x}_0, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by OGM-m for $1 \le m \le N-1$. Then, for any $N \ge 1$,*

$$\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \le ||\nabla f(\boldsymbol{x}_N)|| \le \frac{\sqrt{2}LR}{(m+1)\sqrt{N-m+1}}. \tag{3.16}$$

Proof. Using (3.5), (3.7), (3.11) yields

$$\begin{aligned} \frac{LR^2}{2\vartheta_m^2} &\overset{(3.5)}{\ge} f(\boldsymbol{x}_m) - f(\boldsymbol{x}_*) \overset{(3.7)}{\ge} f(\boldsymbol{x}_{N+1}) - f(\boldsymbol{x}_*) + \frac{1}{2L}\sum_{i=m}^{N} ||\nabla f(\boldsymbol{x}_i)||^2 \\ &\overset{(3.11)}{\ge} \frac{N-m+1}{2L}||\nabla f(\boldsymbol{x}_N)||^2, \end{aligned}$$

which is equivalent to (3.16) using $\vartheta_m \geq \frac{m+1}{\sqrt{2}}$ that is implied by (3.6). $\square$

The bound (3.16) is minimized at a point close to $m = \lfloor 2N/3 \rfloor$, leading to its (approximately) smallest constant $\frac{3\sqrt{6}}{2}$ with the rate $O(1/N^{1.5})$.

Other variants of FGM having $O(1/N^{1.5})$ worst-case gradient bounds were derived in [11, 21]. Such variations of FGM (including FGM-m) were derived since, prior to this paper, it was unknown whether or not FGM decreases the gradient norm with the rate $O(1/N^{1.5})$; this rate for the gradient norm of FGM was conjectured numerically in [31]. Section 5.2 below uses PEP to show for the first time the rate $O(1/N^{1.5})$ for the gradient decrease of the FGM. The bound (3.16) of OGM-m for decreasing the gradient is smaller than the bounds for the FGM variants in [11, 21], and section 5.4 below shows that our proposed methods have worst-case bounds even lower than (3.16).

The preceding sections have focused on tight or upper worst-case bounds of the gradient norm decrease of first-order methods, whereas the next section reviews a lower bound for the worst-case gradient norm decrease in [23], illustrating the best achievable worst-case rate of the gradient norm decrease for any first-order method (with either fixed-step or adaptive-step approaches).

3.2.3. A lower bound of the worst-case rates of first-order methods for decreasing the gradient norm. For completeness, this section reviews a lower bound on the worst-case rate of any first-order method in terms of the gradient norm values for smooth convex *quadratic* functions [23]. Lower bounds on the function value were studied for convex *quadratic* functions in [23] and for smooth convex functions in [7, 25].

When the condition $d \geq 2N+3$ holds, a worst-case gradient norm bound of any first-order method generating $\boldsymbol{x}_N$ after N iterations has rate $O(1/N^2)$ at best, for convex quadratic f, i.e., has the following lower bound [23, sect. 2.3.B]:

$$\frac{LR}{4e^2(N+1)^2} \leq \max_{\substack{f \in \mathcal{Q}_L(\mathbb{R}^d),\\ \boldsymbol{x}_* \in X_*(f),\\ ||\boldsymbol{x}_0 - \boldsymbol{x}_*|| \leq R}} ||\nabla f(\boldsymbol{x}_N)||, \tag{3.17}$$

where $\mathcal{Q}_L(\mathbb{R}^d) := \left\{ f \, : \, f(\boldsymbol{x}) \equiv \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{p}^\top \boldsymbol{x} + \boldsymbol{r} \text{ for } \boldsymbol{x} \in \mathbb{R}^d, \; \boldsymbol{Q} \succeq \boldsymbol{0}, \; ||\boldsymbol{Q}|| \leq L \right\}$. Since $\mathcal{Q}_L(\mathbb{R}^d) \subset \mathcal{F}_L(\mathbb{R}^d)$, the lower bound (3.17) for convex quadratic functions also applies to smooth convex functions.

A regularization technique in [27] achieves the rate $O(1/N^2)$ up to a logarithmic factor. However, its adaptive step coefficients require knowing R in advance, which is undesirable in practice. To the best of our knowledge, whether there exists any FSFOM satisfying such a rate is an open question. Instead, this paper discusses a way to develop FSFOM that achieves an $O(1/N^{1.5})$ gradient norm bound with the smallest constant among known FSFOM.

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

This section reviews a relaxation of the cost function form of PEP [9] and reviews how [9, 15, 16] optimized the step coefficients of the FSFOM class over the cost function form of PEP, leading to OGM. Then we propose a parameterized family of algorithms that generalizes OGM, and we analyze the worst-case cost function decrease of the generalized OGM family.

4.1. Review: Relaxation for the cost function form of PEP. The worst-case bound on the cost function for an FSFOM having given step coefficients $\boldsymbol{h} :=$

$\{h_{i+1,k}\}$ corresponds to a solution of the following PEP problem [9, Prob. (P)]:

$$\text{(P)}\quad \mathcal{B}_{\mathrm{P}}(\boldsymbol{h}, N, d, L, R) := \max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_0,\ldots,\boldsymbol{x}_N\in\mathbb{R}^d,\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*)$$

$$\text{s.t. } \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k}\nabla f(\boldsymbol{x}_k), \quad i = 0,\ldots,N-1.$$

Since problem (P) is impractical to solve due to its functional constraint $f \in \mathcal{F}_L(\mathbb{R}^d)$, [9] relaxed it by the following finite set of inequalities satisfied by f [25, Thm. 2.1.5]:

$$\frac{1}{2L}||\nabla f(\boldsymbol{x}_i) - \nabla f(\boldsymbol{x}_j)||^2 \le f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) - \langle\nabla f(\boldsymbol{x}_j),\, \boldsymbol{x}_i - \boldsymbol{x}_j\rangle \tag{4.1}$$

for $i, j = 0, 1, \ldots, N, *$. Then a matrix $\boldsymbol{G} = [\boldsymbol{g}_0, \ldots, \boldsymbol{g}_N]^\top \in \mathbb{R}^{(N+1)\times d}$ and a vector $\boldsymbol{\delta} = [\delta_0, \ldots, \delta_N]^\top \in \mathbb{R}^{N+1}$ with

$$\boldsymbol{g}_i := \frac{1}{L||\boldsymbol{x}_0 - \boldsymbol{x}_*||}\nabla f(\boldsymbol{x}_i) \quad\text{and}\quad \delta_i := \frac{1}{L||\boldsymbol{x}_0 - \boldsymbol{x}_*||^2}(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_*))$$

are introduced to represent gradient vectors and function values, respectively, in the set of (4.1). This leads to a finite-dimensional relaxation of problem (P) [9, Prob. (Q)]:

$$\text{(P1)}\quad \mathcal{B}_{\mathrm{P1}}(\boldsymbol{h}, N, d, L, R) := \max_{\substack{\boldsymbol{G}\in\mathbb{R}^{(N+1)\times d},\\ \boldsymbol{\delta}\in\mathbb{R}^{N+1}}} LR^2\delta_N$$

$$\begin{aligned} \text{s.t. } & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{A}_{i,j}(\boldsymbol{h})\boldsymbol{G}\} \le \delta_i - \delta_j, \quad i < j = 0,\ldots,N,\\ & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{B}_{i,j}(\boldsymbol{h})\boldsymbol{G}\} \le \delta_i - \delta_j, \quad j < i = 0,\ldots,N,\\ & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{C}_i\boldsymbol{G}\} \le \delta_i, \quad i = 0,\ldots,N,\\ & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{D}_i(\boldsymbol{h})\boldsymbol{G} + \boldsymbol{\nu}\boldsymbol{u}_i^\top\boldsymbol{G}\} \le -\delta_i, \quad i = 0,\ldots,N, \end{aligned}$$

for any given unit vector $\boldsymbol{\nu} \in \mathbb{R}^d$, where $\boldsymbol{u}_i = \boldsymbol{e}_{i+1} \in \mathbb{R}^{N+1}$ is the $(i+1)$th standard basis vector. Note that $\mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{u}_i\boldsymbol{u}_j^\top\boldsymbol{G}\} = \langle\boldsymbol{g}_i,\, \boldsymbol{g}_j\rangle$ by definition. The matrices $\boldsymbol{A}_{i,j}(\boldsymbol{h}), \boldsymbol{B}_{i,j}(\boldsymbol{h}), \boldsymbol{C}_i, \boldsymbol{D}_i(\boldsymbol{h})$ are defined as

$$\begin{cases} \boldsymbol{A}_{i,j}(\boldsymbol{h}) := \frac{1}{2}(\boldsymbol{u}_i - \boldsymbol{u}_j)(\boldsymbol{u}_i - \boldsymbol{u}_j)^\top + \frac{1}{2}\sum_{l=i+1}^{j}\sum_{k=0}^{l-1} h_{l,k}(\boldsymbol{u}_j\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_j^\top),\\ \boldsymbol{B}_{i,j}(\boldsymbol{h}) := \frac{1}{2}(\boldsymbol{u}_i - \boldsymbol{u}_j)(\boldsymbol{u}_i - \boldsymbol{u}_j)^\top - \frac{1}{2}\sum_{l=j+1}^{i}\sum_{k=0}^{l-1} h_{l,k}(\boldsymbol{u}_j\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_j^\top),\\ \boldsymbol{C}_i := \frac{1}{2}\boldsymbol{u}_i\boldsymbol{u}_i^\top,\\ \boldsymbol{D}_i(\boldsymbol{h}) := \frac{1}{2}\boldsymbol{u}_i\boldsymbol{u}_i^\top + \frac{1}{2}\sum_{j=1}^{i}\sum_{k=0}^{j-1} h_{j,k}(\boldsymbol{u}_i\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_i^\top). \end{cases} \tag{4.2}$$

In [9, Prob. (Q′)], problem (P1) is further relaxed by discarding some constraints to yield

$$\text{(P2)}\quad \mathcal{B}_{\mathrm{P2}}(\boldsymbol{h}, N, d, L, R) := \max_{\substack{\boldsymbol{G}\in\mathbb{R}^{(N+1)\times d},\\ \boldsymbol{\delta}\in\mathbb{R}^{N+1}}} LR^2\delta_N$$

$$\begin{aligned} \text{s.t. } & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{A}_{i-1,i}(\boldsymbol{h})\boldsymbol{G}\} \le \delta_{i-1} - \delta_i, \quad i = 1,\ldots,N,\\ & \mathsf{Tr}\{\boldsymbol{G}^\top\boldsymbol{D}_i(\boldsymbol{h})\boldsymbol{G} + \boldsymbol{\nu}\boldsymbol{u}_i^\top\boldsymbol{G}\} \le -\delta_i, \quad i = 0,\ldots,N, \end{aligned}$$

for any given unit vector $\boldsymbol{\nu} \in \mathbb{R}^d$. We explicitly illustrate the relaxation from (P1) to (P2) because section 5 uses a similar but different relaxation. Taylor, Hendrickx, and Glineur [31] avoided this step to analyze a tight worst-case bound of (P) (under a large-scale condition $d \geq N+2$ [31, Thm. 5]); however, the relaxation (P2) facilitates the analysis in [9, 15, 16] and in this paper.

Replacing $\max_{\boldsymbol{G},\boldsymbol{\delta}} LR^2\delta_N$ by $\min_{\boldsymbol{G},\boldsymbol{\delta}}\{-\delta_N\}$ in (P2) for convenience, the Lagrangian of the corresponding constrained minimization problem with dual variables $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^\top \in \mathbb{R}^N$ and $\boldsymbol{\tau} = (\tau_0, \ldots, \tau_N)^\top \in \mathbb{R}^{N+1}$ for the first and second constraint inequalities of (P2), respectively, becomes

$$\mathcal{L}(\boldsymbol{G}, \boldsymbol{\delta}, \boldsymbol{\lambda}, \boldsymbol{\tau}; \boldsymbol{h}) = -\,\delta_N + \sum_{i=1}^{N} \lambda_i(\delta_i - \delta_{i-1}) + \sum_{i=0}^{N} \tau_i \delta_i + \mathsf{Tr}\left\{\boldsymbol{G}^\top \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau})\boldsymbol{G} + \boldsymbol{\nu}\boldsymbol{\tau}^\top \boldsymbol{G}\right\}, \tag{4.3}$$

where

$$\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) := \sum_{i=1}^{N} \lambda_i \boldsymbol{A}_{i-1,i}(\boldsymbol{h}) + \sum_{i=0}^{N} \tau_i \boldsymbol{D}_i(\boldsymbol{h}). \tag{4.4}$$

Then we have the following dual problem of (P2), which one could use to compute a valid upper bound of (P) using a semidefinite program (SDP) for given $\boldsymbol{h}$ [9, Prob. (DQ′)]:

$$\mathcal{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R) := \min_{\substack{(\boldsymbol{\lambda},\boldsymbol{\tau})\in\Lambda,\\ \gamma\in\mathbb{R}}} \left\{ \frac{1}{2}LR^2\gamma \;:\; \begin{pmatrix} \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq \mathbf{0} \right\}, \tag{D}$$

where

$$\Lambda = \left\{ (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \mathbb{R}_+^{2N+1} \;:\; \begin{array}{l} \tau_0 = \lambda_1, \;\; \lambda_N + \tau_N = 1, \\ \lambda_i - \lambda_{i+1} + \tau_i = 0, \; i = 1, \ldots, N-1 \end{array} \right\}. \tag{4.5}$$

The next section reviews the analytical solution to this upper bound (D) for OGM, instead of using a numerical SDP solver.

4.2. Review: Optimizing step coefficients for the cost function form of PEP.

Drori and Teboulle [9] optimized numerically the step coefficients $\boldsymbol{h}$ over the simple SDP problem (D) as follows [9, Prob. (BIL)]:

$$\hat{\boldsymbol{h}}_{\mathrm{D}} := \mathop{\arg\min}_{\boldsymbol{h}\in\mathbb{R}^{N(N+1)/2}} \mathcal{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R). \tag{HD}$$

The problem (HD) is bilinear, and [9, Thm. 3][4] used a convex relaxation technique to make it solvable by numerical methods.

In [15, Lemma 4], we solved (HD) analytically yielding the optimized step coefficients

$$h_{i+1,k} = \begin{cases} \frac{1}{\theta_{i+1}}\left(2\theta_k - \sum_{j=k+1}^{i} h_{j,k}\right), & k = 0, \ldots, i-1, \\ 1 + \frac{2\theta_i - 1}{\theta_{i+1}}, & k = i, \end{cases} \tag{4.6}$$

[4] [9, Thm. 3] has typos that are fixed in [15, eqn. (6.3)].

for θ_i in (3.6). Fortuitously, the optimized coefficients (4.6) lead to equivalent computationally efficient OGM1 and OGM2 forms [15, Props. 3, 4, and 5], and the bound (3.5) for the final secondary iterate $\boldsymbol{x}_N$ of OGM is implied by [15, Lem. 4]. Recently, Drori [7] showed that the OGM is optimal for $d \ge N+1$, implying that optimizing over the relaxed bound (D) in (HD) for simplicity is equivalent to optimizing over the exact worst-case cost function bound (P) when $d \ge N+1$.

One could use an SDP solver to compute a numerical bound from (D) for any FSFOM; however, deriving an analytical bound using (D) is difficult for the primary sequence $\{\boldsymbol{y}_i\}$ of OGM. Therefore, we devised a new relaxed bound in [16] similar to (D), which we review next.

4.3. Review: Another cost function form of relaxed PEP for the primary sequence of OGM. An upper bound of the worst-case bound on $f(\boldsymbol{y}_{N+1}) - f(\boldsymbol{x}_*)$ for FSFOM with step coefficients $\boldsymbol{h}$ and $\boldsymbol{y}_{N+1} = \boldsymbol{x}_N - \frac{1}{L} f(\boldsymbol{x}_N)$ could be computed using (D) by an SDP solver. However, we found it difficult to find its analytical worst-case bound for the primary sequence $\{\boldsymbol{y}_i\}$ of OGM, so [16, Prob. (D′)] provided the following alternate upper bound on $f(\boldsymbol{y}_{N+1}) - f(\boldsymbol{x}_*)$:

(D′)

$$\mathcal{B}_{\mathrm{D}'}(\boldsymbol{h}, N, L, R) := \min_{\substack{(\boldsymbol{\lambda},\boldsymbol{\tau})\in\Lambda,\\ \gamma\in\mathbb{R}}} \left\{ \frac{1}{2} L R^2 \gamma \;:\; \begin{pmatrix} \boldsymbol{S}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau}) + \frac{1}{2}\boldsymbol{u}_N \boldsymbol{u}_N^\top & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq \boldsymbol{0} \right\},$$

which led to the bound (3.4) for the primary sequence $\{\boldsymbol{y}_i\}$ of OGM in [16].

Similar to [15, Lemma 4], we found a feasible point of (D′) in [16, Lemma 3.1], along with feasible step coefficients $\boldsymbol{h}$ of an FSFOM:

$$h_{i+1,k} = \begin{cases} \frac{1}{t_{i+1}} \left(2t_k - \sum_{j=k+1}^{i} h_{j,k} \right), & k = 0, \dots, i-1, \\ 1 + \frac{2t_i - 1}{t_{i+1}}, & k = i, \end{cases} \tag{4.7}$$

for t_i in (3.3). Then [16, Thm. 3.1] showed the bound (3.4) using [16, Lem. 3.1]. The step coefficients (4.6) and (4.7) are identical except for the final iteration, since t_i (3.3) and θ_i (3.6) are equivalent for $i < N$.

We are now done reviewing the portions of papers [15, 16] that are the ingredients for specifying a parameterized family of algorithms that generalizes OGM in the next two sections.

4.4. Feasible points of (D) and (D′) for the generalized OGM. This section specifies feasible points of (D) and (D′) that lead to a generalized version of OGM. Specifically, the following lemma presents additional feasible points of (D); this lemma reduces to [15, Lem. 4] (and the step coefficients (4.6) of OGM) when $\theta_i^2 = \Omega_i$ for all i.

LEMMA 4.1. *For the step coefficients*

$$h_{i+1,k} = \begin{cases} \frac{\theta_{i+1}}{\Omega_{i+1}} \left(2\theta_k - \sum_{j=k+1}^{i} h_{j,k} \right), & i = 0, \dots, N-1,\ k = 0, \dots, i-1, \\ 1 + \frac{(2\theta_i - 1)\theta_{i+1}}{\Omega_{i+1}}, & i = 0, \dots, N-1,\ k = i, \end{cases} \tag{4.8}$$

the choice of variables

$$\gamma = \frac{1}{2}\tau_0, \quad \lambda_i = \Omega_{i-1}\tau_0, \; i = 1, \dots, N, \quad \tau_i = \begin{cases} \frac{2}{\Omega_N}, & i = 0, \\ \theta_i\tau_0 & i = 1, \dots, N-1, \\ \frac{\theta_N}{2}\tau_0, & i = N, \end{cases} \tag{4.9}$$

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

$$\theta_0 = 1, \quad \theta_i > 0, \quad \textit{and} \quad \theta_i^2 \le \Omega_i := \begin{cases} \sum_{l=0}^{i}\theta_l, & i = 0, \dots, N-1, \\ 2\sum_{l=0}^{N-1}\theta_l + \theta_N, & i = N. \end{cases} \tag{4.10}$$

Proof. See Appendix B. □

The following lemma also specifies some feasible points of (D′); this lemma reduces to [16, Lem. 3.1] (and (4.7)) when $t_i^2 = T_i$ for all i.

LEMMA 4.2. *For the step coefficients*

$$h_{i+1,k} = \begin{cases} \frac{t_{i+1}}{T_{i+1}}\left(2t_k - \sum_{j=k+1}^{i} h_{j,k}\right), & i = 0, \dots, N-1, \; k = 0, \dots, i-1, \\ 1 + \frac{(2t_i - 1)t_{i+1}}{T_{i+1}}, & i = 0, \dots, N-1, \; k = i, \end{cases} \tag{4.11}$$

the choice of variables

$$\gamma = \frac{1}{2}\tau_0, \quad \lambda_i = T_{i-1}\tau_0, \; i = 1, \dots, N, \quad \tau_i = \begin{cases} \frac{1}{T_N}, & i = 0, \\ t_i\tau_0 & i = 1, \dots, N, \end{cases} \tag{4.12}$$

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

$$t_0 = 1, \quad t_i > 0, \quad \textit{and} \quad t_i^2 \le T_i := \sum_{l=0}^{i} t_l. \tag{4.13}$$

Proof. See Appendix C. □

Similar to the relationship between the step coefficients (4.6) and (4.7), the step coefficients (4.8) and (4.11) are identical (when $\theta_i = t_i$ for $i < N$) except for the final iteration, implying that the iterates $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ of the two FSFOMs with (4.8) and (4.11) are equivalent; only the final iterate $\boldsymbol{x}_N$ is different. The step coefficients (4.6) and (4.7) lead to computationally efficient equivalent OGM forms; similarly the next section provides computationally efficient generalized forms of OGM that each correspond to a FSFOM with either (4.8) or (4.11), and we analyze their cost function worst-case bounds.

4.5. Generalized OGM. This section proposes a generalized OGM using Lemmas 4.1 and 4.2. The FSFOM with the step coefficients (4.8) has the following two equivalent efficient generalized forms of OGM, named GOGM1 and GOGM2, that reduce to the standard OGM when $\theta_i^2 = \Omega_i$ for all i.

```
Algorithm GOGM1
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $\theta_0 = \Omega_0 = 1$.
For $i = 0, \dots, N-1$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    Choose $\theta_{i+1} > 0$
    s.t. $\theta_{i+1}^2 \le \Omega_{i+1}$, where
    $\Omega_{i+1} = \begin{cases} \sum_{l=0}^{i+1} \theta_l, & i \le N-2 \\ 2\sum_{l=0}^{N-1} \theta_l + \theta_N, & i = N-1 \end{cases}$
    $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1} + \frac{(\Omega_i - \theta_i)\theta_{i+1}}{\theta_i \Omega_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i)$
            $+ \frac{(2\theta_i^2 - \Omega_i)\theta_{i+1}}{\theta_i \Omega_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i)$
```

```
Algorithm GOGM2
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $\theta_0 = \Omega_0 = 1$.
For $i = 0, \dots, N-1$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    $\boldsymbol{z}_{i+1} = \boldsymbol{x}_0 - \frac{1}{L}\sum_{k=0}^{i} 2\theta_k \nabla f(\boldsymbol{x}_k)$
    Choose $\theta_{i+1} > 0$
    s.t. $\theta_{i+1}^2 \le \Omega_{i+1}$, where
    $\Omega_{i+1} = \begin{cases} \sum_{l=0}^{i+1} \theta_l, & i \le N-2 \\ 2\sum_{l=0}^{N-1} \theta_l + \theta_N, & i = N-1 \end{cases}$
    $\boldsymbol{x}_{i+1} = \left(1 - \frac{\theta_{i+1}}{\Omega_{i+1}}\right)\boldsymbol{y}_{i+1} + \frac{\theta_{i+1}}{\Omega_{i+1}}\boldsymbol{z}_{i+1}$
```

Proposition 4.3. *The sequence $\{\boldsymbol{x}_0, \dots, \boldsymbol{x}_N\}$ generated by the FSFOM with* (4.8) *is identical to the corresponding sequence generated by GOGM*1 *and GOGM*2.

Proof. See Appendix D. Note that this proof is independent of the choice of θ_i and Ω_i. ☐

Because the proof of Proposition 4.3 for the FSFOM with step coefficients (4.8) is independent of the choice of θ_i and Ω_i, it is straightforward to show that the FSFOM with step coefficients (4.11) has the following two efficient equivalent forms, named GOGM1′ and GOGM2′, that reduce to [16, Algs. OGM1′ and OGM2′] when $t_i^2 = T_i$ for all i.

```
Algorithm GOGM1′
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $t_0 = T_0 = 1$.
For $i = 0, \dots, N-1$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    Choose $t_{i+1} > 0$
    s.t. $t_{i+1}^2 \le T_{i+1} = \sum_{l=0}^{i+1} t_l$
    $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1} + \frac{(T_i - t_i)t_{i+1}}{t_i T_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i)$
            $+ \frac{(2t_i^2 - T_i)t_{i+1}}{t_i T_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i)$
```

```
Algorithm GOGM2′
Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 = \boldsymbol{y}_0 \in \mathbb{R}^d$,
        $t_0 = T_0 = 1$.
For $i = 0, \dots, N-1$
    $\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\nabla f(\boldsymbol{x}_i)$
    $\boldsymbol{z}_{i+1} = \boldsymbol{x}_0 - \frac{1}{L}\sum_{k=0}^{i} 2t_k \nabla f(\boldsymbol{x}_k)$
    Choose $t_{i+1} > 0$
    s.t. $t_{i+1}^2 \le T_{i+1} = \sum_{l=0}^{i+1} t_l$
    $\boldsymbol{x}_{i+1} = \left(1 - \frac{t_{i+1}}{T_{i+1}}\right)\boldsymbol{y}_{i+1} + \frac{t_{i+1}}{T_{i+1}}\boldsymbol{z}_{i+1}$
```

Clearly when $\theta_i = t_i$ for $i < N$, the primary iterates $\{\boldsymbol{y}_i\}_{i=0}^N$ and the intermediate secondary iterates $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ of GOGM and GOGM′ are equivalent. Although illustrating two similar algorithms GOGM and GOGM′ might seem redundant, presenting both formulations with Lemmas 4.1 and 4.2 completes the story of generalized OGM here and in section 5.

Using Lemmas 4.1 and 4.2, the following theorem bounds the cost function decrease of the GOGM and GOGM′ iterates.

Theorem 4.4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{y}_0, \dots, \boldsymbol{y}_N, \boldsymbol{x}_N \in \mathbb{R}^d$ be gener-*

ated by GOGM1 and GOGM2. Then, for any $1 \le i \le N$,

$$f(\boldsymbol{y}_i) - f(\boldsymbol{x}_*) \le \frac{LR^2}{4\Omega_{i-1}}, \tag{4.14}$$

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*) \le \frac{LR^2}{2\Omega_N}. \tag{4.15}$$

The iterates $\boldsymbol{y}_0, \dots, \boldsymbol{y}_N \in \mathbb{R}^d$ generated by GOGM1′ and GOGM2′ also satisfy the bound (4.14) when $\theta_i = t_i$ for $i < N$.

Proof. Using Lemma 4.2, the FSFOM with $\boldsymbol{h}$ in (4.11) of GOGM′ satisfies

$$f(\boldsymbol{y}_{N+1}) - f(\boldsymbol{x}_*) \le \mathcal{B}_{\mathrm{D}'}(\boldsymbol{h}, N, L, R) = \frac{1}{2}LR^2\gamma = \frac{LR^2}{4T_N}, \tag{4.16}$$

where $\boldsymbol{y}_{N+1} = \boldsymbol{x}_N - \frac{1}{L}\nabla f(\boldsymbol{x}_N)$. Since the coefficients $\boldsymbol{h}$ in (4.11) are recursive and do not depend on a given N, we can extend (4.16) for all iterations. By letting $\theta_i = t_i$ for $i < N$, the bound (4.16) also satisfies for the iterates $\{\boldsymbol{y}_i\}$ of GOGM, as in (4.14).

Using Lemma 4.1, the FSFOM with the step $\boldsymbol{h}$ in (4.8) of GOGM satisfies

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*) \le \mathcal{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R) = \frac{1}{2}LR^2\gamma = \frac{LR^2}{2\Omega_N}, \tag{4.17}$$

which is equivalent to (4.15). □

GOGM and Theorem 4.4 reduce to OGM and its bounds (3.4) and (3.5) when $\theta_i^2 = \Omega_i$ for all i. Similar to general forms of FGM in [3, 26], the GOGM family includes the choice

$$\theta_i = \begin{cases} \frac{i+a}{a}, & i < N, \\ \frac{\sqrt{2}(N+a-1)}{a}, & i = N, \end{cases}$$

for any $a \ge 2$, because such a parameter θ_i satisfies the following conditions for GOGM:

$$\Omega_i - \theta_i^2 = \frac{(i+1)(i+2a)}{2a} - \frac{(i+a)^2}{a^2} = \frac{(a-2)i^2 + a(2a-3)}{2a^2} \ge 0 \tag{4.18}$$

for $i < N$, and $\Omega_N - \theta_N^2 = 2\Omega_{N-1} + \theta_N - \theta_N^2 \ge 2\theta_{N-1}^2 + \theta_N - \theta_N^2 = \theta_N \ge 0$. Similarly, the GOGM′ family includes the choice $t_i = \frac{i+a}{a}$ for any $a \ge 2$, which we denote by OGM-a.

Corollary 4.5. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{y}_0, \dots, \boldsymbol{y}_N \in \mathbb{R}^d$ be generated by GOGM′ with $t_i = \frac{i+a}{a}$ (OGM-a) for any $a \ge 2$. Then, for any $1 \le i \le N$,*

$$f(\boldsymbol{y}_i) - f(\boldsymbol{x}_*) \le \frac{aLR^2}{2i(i+2a-1)}. \tag{4.19}$$

Proof. Theorem 4.4 implies (4.19), since $T_i = \frac{(i+1)(i+2a)}{2a}$ and the condition $T_i - t_i^2 \ge 0$ satisfies in (4.18) for any $a \ge 2$. □

5. Relaxation and optimization of the gradient form of PEP. This section analyzes a worst-case bound for the gradient of any GOGM (and GOGM′) using the gradient form of PEP. We use relaxations on the gradient form of PEP that are similar but slightly different from those of PEP for the cost function in the previous section. Using this relaxed PEP, we prove that FGM has an $O(1/N^{1.5})$ rate for the worst-case gradient decrease and analyze the worst-case gradient bound for the GOGM. Then we optimize the step coefficients with respect to the gradient form of PEP and propose an algorithm named OGM-OG that lies in the GOGM family and that has the best known analytical worst-case bound for decreasing the gradient norm among the FSFOM class.

5.1. Relaxation for the gradient form of PEP.

To analyze a worst-case bound on the gradient for an FSFOM with a given $\boldsymbol{h}$, we consider the following gradient-form version of PEP that is similar to (P):

$$\text{(P}''\text{)}\quad \mathcal{B}_{\text{P}''}(\boldsymbol{h},N,d,L,R) := \max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_0,\ldots,\boldsymbol{x}_N\in\mathbb{R}^d,\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} \min_{i\in\{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)||^2$$

$$\text{s.t. } \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k}\nabla f(\boldsymbol{x}_k), \quad i=0,\ldots,N-1.$$

Here, we use the smallest gradient norm squared among all iterates ($\min_{i\in\{0,\ldots,N\}}$ $||\nabla f(\boldsymbol{x}_i)||^2$) as a criteria, as considered in [31, sect. 4.3]. We could instead consider the final gradient norm squared ($||\nabla f(\boldsymbol{x}_N)||^2$) as a criteria, but our proposed relaxation on (P$''$) in this section for such criteria provided only an $O(1/N)$ worst-case bound at best even for the corresponding optimized step coefficients (results not shown); we leave studying the gradient form of the tight PEP as future work.

As in [31], we replace $\min_{i\in\{0,\ldots,N\}}||\nabla f(\boldsymbol{x}_i)||^2$ in (P$''$) by $L^2||\boldsymbol{x}_0-\boldsymbol{x}_*||^2\alpha$ with the condition $\alpha \le \frac{1}{L^2||\boldsymbol{x}_0-\boldsymbol{x}_*||^2}||\nabla f(\boldsymbol{x}_i)||^2 = \operatorname{Tr}\{\boldsymbol{G}^\top(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\boldsymbol{G}\}$ for all i. Then we relax this reformulated (P$''$) similarly to the relaxation from (P) to (P2) with the additional constraint $\operatorname{Tr}\{\boldsymbol{G}^\top\boldsymbol{C}_N\boldsymbol{G}\}\le\delta_N$ in (P1) as follows:

$$\text{(P2}''\text{)}\quad \mathcal{B}_{\text{P2}''}(\boldsymbol{h},N,d,L,R) := \max_{\substack{\boldsymbol{G}\in\mathbb{R}^{(N+1)d},\\ \boldsymbol{\delta}\in\mathbb{R}^{N+1},\\ \alpha\in\mathbb{R}}} L^2R^2\alpha$$

$$\begin{aligned}\text{s.t. } & \operatorname{Tr}\{\boldsymbol{G}^\top\boldsymbol{A}_{i-1,i}(\boldsymbol{h})\boldsymbol{G}\}\le\delta_{i-1}-\delta_i, \quad i=1,\ldots,N,\\ & \operatorname{Tr}\{\boldsymbol{G}^\top\boldsymbol{C}_N\boldsymbol{G}\}\le\delta_N,\\ & \operatorname{Tr}\{\boldsymbol{G}^\top\boldsymbol{D}_i(\boldsymbol{h})\boldsymbol{G}+\boldsymbol{\nu}\boldsymbol{u}_i^\top\boldsymbol{G}\}\le-\delta_i, \quad i=0,\ldots,N,\\ & \operatorname{Tr}\{\boldsymbol{G}^\top(-\boldsymbol{u}_i\boldsymbol{u}_i^\top)\boldsymbol{G}\}\le-\alpha, \quad i=0,\ldots,N.\end{aligned}$$

Replacing $\max_{\boldsymbol{G},\boldsymbol{\delta}} L^2R^2\alpha$ by $\min_{\boldsymbol{G},\boldsymbol{\delta}}\{-\alpha\}$ for convenience, the Lagrangian of the corresponding constrained minimization problem with dual variables $\boldsymbol{\lambda}\in\mathbb{R}_+^N$, $\eta\in\mathbb{R}_+$, $\boldsymbol{\tau}\in\mathbb{R}_+^{N+1}$, and $\boldsymbol{\beta}=(\beta_0,\ldots,\beta_N)^\top\in\mathbb{R}_+^{N+1}$ for the first, second, third, and fourth set of constraint inequalities of (P2$''$), respectively, becomes

$$\text{(5.1)}\quad \begin{aligned}\mathcal{L}''(\boldsymbol{G},\boldsymbol{\delta},\boldsymbol{\lambda},\eta,\boldsymbol{\tau},\boldsymbol{\beta};\boldsymbol{h}) = & -\alpha+\sum_{i=1}^N\lambda_i(\delta_i-\delta_{i-1})-\eta\delta_N+\sum_{i=0}^N\tau_i\delta_i+\sum_{i=0}^N\beta_i\alpha\\ & +\operatorname{Tr}\{\boldsymbol{G}^\top\boldsymbol{S}''(\boldsymbol{h},\boldsymbol{\lambda},\eta,\boldsymbol{\tau},\boldsymbol{\beta})\boldsymbol{G}+\boldsymbol{\nu}\boldsymbol{\tau}^\top\boldsymbol{G}\},\end{aligned}$$

where

$$\boldsymbol{S}''(\boldsymbol{h},\boldsymbol{\lambda},\eta,\boldsymbol{\tau},\boldsymbol{\beta}) := \boldsymbol{S}(\boldsymbol{h},\boldsymbol{\lambda},\boldsymbol{\tau}) + \frac{1}{2}\eta\boldsymbol{u}_N\boldsymbol{u}_N^\top - \sum_{i=0}^N \beta_i\boldsymbol{u}_i\boldsymbol{u}_i^\top. \tag{5.2}$$

Then, similarly to (D), we have the following dual problem of (P2″), which one could use to compute an upper bound of the PEP (P″) of the smallest gradient norm squared among all iterates by a numerical SDP solver:

(D″)

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

where

$$\Lambda'' := \left\{(\boldsymbol{\lambda},\eta,\boldsymbol{\tau},\boldsymbol{\beta}) \in \mathbb{R}_+^{3N+3} \;:\; \begin{array}{l} \tau_0 = \lambda_1,\;\; \lambda_N + \tau_N = \eta,\;\; \sum_{i=0}^N \beta_i = 1, \\ \lambda_i - \lambda_{i+1} + \tau_i = 0,\; i = 1,\dots,N-1 \end{array}\right\}. \tag{5.3}$$

The next two sections use a valid upper bound (D″) of (P″) for given step coefficients $\boldsymbol{h}$, providing an analytical solution to (D″) for the step coefficients $\boldsymbol{h}$ of FGM and GOGM′, superseding the use of an SDP solver.

5.2. A worst-case bound for the gradient norm of FGM. FGM is equivalent to an FSFOM with the step coefficients [15, Prop. 1]:

$$h_{i+1,k} = \begin{cases} \frac{1}{t_{i+1}}\left(t_k - \sum_{j=k+1}^i h_{j,k}\right), & k = 0,\dots,i-1, \\ 1 + \frac{t_i - 1}{t_{i+1}}, & k = i, \end{cases} \tag{5.4}$$

for t_i in (3.3). The following lemma provides a feasible point of (D″) associated with the step coefficients (5.4) of FGM to provide a worst-case bound for the gradient of FGM.

LEMMA 5.1. *For the step coefficients* (5.4)*, the choice of variables*

(5.5)

$$\gamma = \tau_0, \quad \lambda_i = t_{i-1}^2\tau_0, \quad i = 1,\dots,N, \quad \tau_i = \begin{cases} \left(\frac{1}{2}\sum_{k=0}^N t_k^2\right)^{-1}, & i = 0, \\ t_i\tau_0, & i = 1,\dots,N, \end{cases}$$

(5.6)

$$\eta = t_N^2\tau_0, \quad \beta_i = \frac{1}{2}t_i^2\tau_0, \quad i = 0,\dots,N,$$

is a feasible point of (D″) *for t_i in* (3.3).

Proof. See Appendix E. □

Using Lemma 5.1, the following theorem bounds the gradient norm of the FGM iterates, proving for the first time an $O(1/N^{1.5})$ rate of decrease.

THEOREM 5.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{y}_0,\dots,\boldsymbol{y}_N,\boldsymbol{x}_0,\dots,\boldsymbol{x}_N \in \mathbb{R}^d$ be generated by FGM. Then, for any $N \geq 1$,*

$$\begin{aligned} \min_{i\in\{0,\dots,N+1\}} ||\nabla f(\boldsymbol{y}_i)|| &\leq \min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \\ &\leq \frac{LR}{\sqrt{\sum_{k=0}^N t_k^2}} \leq \frac{2\sqrt{3}LR}{\sqrt{(N+1)(N^2+6N+12)}}, \end{aligned} \tag{5.7}$$

where $\boldsymbol{y}_{N+1} = \boldsymbol{x}_N - \frac{1}{L}\nabla f(\boldsymbol{x}_N)$.

Proof. Lemma 3.1 implies the first inequality in (5.7). Using Lemma 5.1, the FSFOM with the step coefficients $\boldsymbol{h}$ (5.4) of FGM satisfies

$$\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)||^2 \le \mathcal{B}_{\mathrm{D}''}(\boldsymbol{h},N,L,R) = \frac{1}{2}L^2R^2\gamma = \frac{L^2R^2}{\sum_{k=0}^N t_k^2}, \tag{5.8}$$

which is equivalent to (5.7) using $\sum_{k=0}^N t_k^2 \ge \sum_{k=0}^N \frac{(k+2)^2}{4} = \frac{(N+1)(2N^2+13N+24)}{24}$. □

5.3. A worst-case bound for the gradient norm of GOGM.

Having established the gradient bound (5.7) for FGM, this section and the next seek to improve on it by studying GOGM. To bound the gradient decrease of GOGM (and GOGM′), the following lemma illustrates one possible set of feasible points of (D″).

Lemma 5.3. *For the step coefficients* (4.11), *the choice of variables*

(5.9)

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

$$\eta = T_N\tau_0, \quad \beta_i = \left(T_i - t_i^2\right)\tau_0, \quad i=0,\dots,N, \tag{5.10}$$

is a feasible point of (D″) *for any choice of* t_i *and* T_i *that satisfies* (4.13) *and for which there exists some* i *such that* $t_i^2 < T_i$.

Proof. See Appendix F. □

Using Lemma 5.3, the following theorem bounds the worst-case gradient norm for the iterates of GOGM and GOGM′.

Theorem 5.4. *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *be* $\mathcal{F}_L(\mathbb{R}^d)$ *and let* $\boldsymbol{y}_0,\dots,\boldsymbol{y}_N,\boldsymbol{x}_0,\dots,\boldsymbol{x}_N \in \mathbb{R}^d$ *be generated by GOGM′. Then, for any* $N \ge 1$,

$$\min_{i\in\{0,\dots,N+1\}} ||\nabla f(\boldsymbol{y}_i)|| \le \min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \le \frac{LR}{2\sqrt{\sum_{k=0}^N \left(T_k - t_k^2\right)}}, \tag{5.11}$$

where $\boldsymbol{y}_{N+1} = \boldsymbol{x}_N - \frac{1}{L}\nabla f(\boldsymbol{x}_N)$. *The bound* (5.11) *can be generalized to the intermediate iterates* $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ *and* $\{\boldsymbol{y}_i\}_{i=0}^N$ *of both GOGM and GOGM′ when* $\theta_i = t_i$ *(for* $i < N$*).*

Proof. Lemma 3.1 implies the first inequality in (5.11). Using Lemma 5.3, FSFOM with the step coefficients $\boldsymbol{h}$ (4.11) of GOGM′ satisfies

$$\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)||^2 \le \mathcal{B}_{\mathrm{D}''}(\boldsymbol{h},N,L,R) = \frac{1}{2}L^2R^2\gamma = \frac{L^2R^2}{4\sum_{k=0}^N \left(T_k - t_k^2\right)},$$

which implies (5.11). Since the iterates of GOGM′ are recursive and do not depend on a given N, the bound (5.11) easily generalizes to the intermediate iterates of GOGM′ (and GOGM when $\theta_i = t_i$). □

5.4. Optimizing step coefficients over the gradient form of PEP.

In search of an FSFOM that decreases the gradient norm the fastest, we optimize the step coefficients in terms of the gradient form of the relaxed (D$''$) by solving the following problem:

$$\text{(HD}''\text{)} \qquad \hat{\boldsymbol{h}}_{\text{D}''} := \operatorname*{arg\,min}_{\boldsymbol{h}\in\mathbb{R}^{N(N+1)/2}} \mathcal{B}_{\text{D}''}(\boldsymbol{h}, N, L, R).$$

The problem (HD$''$) is bilinear, similar to (HD), and a convex relaxation technique [9, Thm. 3] makes this problem solvable using numerical methods.

We solved (HD$''$) for many choices of N using a numerical SDP solver [4, 12] and observed that the following choice of t_i makes the feasible point in Lemma 5.3 optimal for the problem (HD$''$):

$$\text{(5.12)} \qquad t_i = \begin{cases} 1, & i = 0, \\ \frac{1+\sqrt{1+4t_{i-1}^2}}{2}, & i = 1, \dots, \lfloor N/2 \rfloor - 1, \\ \frac{N-i+1}{2}, & i = \lfloor N/2 \rfloor, \dots, N. \end{cases}$$

Based on that numerical evidence, we conjecture that $\hat{\boldsymbol{h}}_{\text{D}''}$ in (HD$''$) corresponds to the step coefficients (4.11) with the parameter t_i (5.12). The t_i factors in (5.12) start decreasing after $i = \lfloor N/2 \rfloor - 1$, whereas the usual t_i in (3.3) and $t_i = \frac{i+a}{a}$ for any $a \geq 2$ increase with i indefinitely.

In addition, we found numerically that minimizing the gradient bound (5.11) of GOGM$'$, i.e., solving the constrained quadratic problem

$$\text{(5.13)} \qquad \max_{\{t_i\}} \sum_{k=0}^{N} \left(\sum_{l=0}^{k} t_l - t_k^2 \right) \quad \text{s.t.} \quad t_i \text{ satisfies (4.13) for all } i,$$

is equivalent to solving the problem (HD$''$). In other words, the solution of (5.13) numerically appears equivalent to (5.12), the (conjectured) solution of (HD$''$). The unconstrained maximizer of the cost function of (5.13) is $t_i = \frac{N-i+1}{2}$, and this term partially appears in the constrained maximizer (5.12) for $\lfloor N/2 \rfloor \leq i \leq N$.

We denote the resulting GOGM$'$ with (5.12) as OGM-OG (OG for "optimized over gradient"). The following theorem bounds the cost function and gradient norm of the OGM-OG iterates.

THEOREM 5.5. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{y}_0, \dots, \boldsymbol{y}_N, \boldsymbol{x}_0, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by OGM-OG. Then*

$$\text{(5.14)} \qquad f(\boldsymbol{y}_{N+1}) - f(\boldsymbol{x}_*) \leq \frac{2LR^2}{(N+2)^2},$$

$$\text{(5.15)} \qquad \min_{i\in\{0,\dots,N+1\}} ||\nabla f(\boldsymbol{y}_i)|| \leq \min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \leq \frac{\sqrt{6}LR}{N\sqrt{N+1}},$$

where $y_{N+1} = \boldsymbol{x}_N - \frac{1}{L}\nabla f(\boldsymbol{x}_N)$.

Proof. OGM-OG is an instance of GOGM$'$, and thus Theorem 4.4 implies that OGM-OG satisfies

$$f(\boldsymbol{y}_{N+1}) - f(\boldsymbol{x}_*) \leq \frac{LR^2}{4T_N},$$

which is equivalent to (5.14), since

$$\begin{aligned} T_N &= T_m + \sum_{l=m+1}^{N} t_l = t_m^2 + \frac{(N-m)(N-m+1)}{4} \\ &\geq \frac{(N+3)^2 + N(N+2)}{16} = \frac{2N^2+8N+9}{16} \end{aligned}$$

for $m = \lfloor N/2 \rfloor$, using $m \geq \frac{N-1}{2}$, $N-m \geq \frac{N}{2}$, and $t_m \geq \frac{m+2}{2} \geq \frac{N+3}{4}$ in (3.3).

Theorem 5.4 implies (5.15), using the above inequalities for m, the equality $t_i^2 = T_i$ for $i \leq m$, and

$$\begin{aligned} &\sum_{k=m+1}^{N} \left(T_k - t_k^2\right) = \sum_{k=m+1}^{N} \left(t_m^2 + \sum_{l=m+1}^{k} t_l - t_k^2 \right) \\ &= (N-m)\, t_m^2 + \sum_{k=m+1}^{N} \left(\sum_{l=m+1}^{k} \frac{N-l+1}{2} - \left(\frac{N-k+1}{2}\right)^2 \right) \\ &= (N-m)\, t_m^2 + \sum_{k'=1}^{N-m} \left(\sum_{l'=1}^{k'} \frac{N-l'-m+1}{2} - \left(\frac{N-k'-m+1}{2}\right)^2 \right) \\ &= (N-m)\, t_m^2 \\ &\quad + \sum_{k=1}^{N-m} \left(\frac{2(N-m+1)k - k(k+1)}{4} - \frac{(N-m+1)^2 - 2(N-m+1)k + k^2}{4} \right) \\ &= (N-m)\, t_m^2 + \sum_{k=1}^{N-m} \left(-\frac{k^2}{2} + (N-m+3/4)k - \frac{(N-m+1)^2}{4} \right) \\ &= (N-m)\, t_m^2 - \frac{(N-m)(N-m+1/2)(N-m+1)}{6} \\ &\quad + \frac{(N-m)(N-m+3/4)(N-m+1)}{2} - \frac{(N-m)(N-m+1)^2}{4} \\ &\geq \frac{(N-m)(m+2)^2}{4} + \frac{(N-m)^2(N-m+1)}{3} - \frac{(N-m)(N-m+1)^2}{4} \\ &\geq \frac{(N-m)^2(N-m+1)}{3} \geq \frac{1}{24} N^2(N+1). \qquad \square \end{aligned}$$

The gradient bound (5.15) of OGM-OG is asymptotically $\sqrt{2}$ times smaller than that of FGM in Theorem 5.2 and 1.5 times smaller than that of OGM-$m = \lfloor 2N/3 \rfloor$ in Theorem 3.5. Regarding the cost function decrease, the bound (5.14) of OGM-OG is asymptotically the same as the bound (3.2) of FGM, and both are two times larger than the bounds (3.4) and (3.5) of OGM.

5.5. Decreasing the gradient norm with rate $O(1/N^{1.5})$ using GOGM without selecting N in advance. Although OGM-OG satisfies a small worst-case gradient bound with a rate $O(1/N^{1.5})$, OGM-OG (and FGM-m and OGM-m) must select N in advance, unlike FGM. Using Theorem 5.4, the following corollary shows that OGM-a with $a > 2$ can decrease the gradient with a rate $O(1/N^{1.5})$ without selecting N in advance. (Corollary 4.5 showed that OGM-a algorithm with $a \geq 2$ can decrease the cost function with an optimal rate $O(1/N^2)$.)

COROLLARY 5.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{F}_L(\mathbb{R}^d)$ and let $\boldsymbol{y}_0, \dots, \boldsymbol{y}_N, \boldsymbol{x}_0, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by GOGM′ with $t_i = \frac{i+a}{a}$ (OGM-a) for any $a \geq 2$. Then, for $N \geq 1$,*

$$\min_{i\in\{0,\dots,N+1\}} ||\nabla f(\boldsymbol{y}_i)|| \leq \min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \leq \frac{a\sqrt{6}LR}{2\sqrt{N(N+1)\left((a-2)N+(3a^2-4a-2)\right)}}, \tag{5.16}$$

where $\boldsymbol{y}_{N+1} = \boldsymbol{x}_N - \frac{1}{L}\nabla f(\boldsymbol{x}_N)$.

Proof. Using $T_i = \frac{(i+1)(i+2a)}{2a}$ and (4.18), Theorem 5.4 implies (5.16) since

$$\sum_{k=0}^{N}(T_k - t_k^2) = \sum_{k=0}^{N} \frac{(a-2)k^2 + a(2a-3)k}{2a^2} = \frac{N(N+1)\big((a-2)N + (3a^2-4a-2)\big)}{6a^2}. \qquad \square$$

OGM-a for any $a > 2$ has a gradient bound (5.16) that is about $\frac{a}{2\sqrt{a-2}}$ times larger than the bound (5.15) of OGM-OG. This constant factor minimizes to $\sqrt{2}$ when $a = 4$, and this OGM-$a = 4$ has a worst-case gradient bound that is asymptotically equivalent to the bound (5.7) of FGM. Therefore, when one does not want to select N in advance, both FGM and OGM-$a = 4$ (and OGM-a for any $a > 2$) will be useful for decreasing the gradient with a rate $O(1/N^{1.5})$.

6. Discussion.

This section summarizes analytical worst-case bounds of FSFOM discussed in the previous sections. This section also reports tight numerical worst-case bounds for exact comparison of algorithms because many of the analytical bounds are not guaranteed to be tight.

6.1. Summary of analytical worst-case bounds on the cost function and gradient norm.

Table 1 summarizes the asymptotic rate of analytical worst-case bounds of all algorithms described in this paper. As discussed, OGM and OGM-OG have the best known worst-case bounds for the cost function and gradient decrease, respectively, in Table 1. However, since OGM has a slow worst-case rate for the gradient decrease, other algorithms such as FGM, OGM-m, OGM-OG, and OGM-a that satisfy both the optimal rate $O(1/N^2)$ for the function decrease and a fast rate $O(1/N^{1.5})$ for the gradient decrease could be preferable over OGM when one is interested in both the gradient decrease as well as the function decrease, particularly when solving dual problems. In addition, when one does not want to choose N in advance, FGM and OGM-a could be preferable.

6.2. Tight worst-case bounds on the cost function and the gradient norm.

Since many worst-case bounds presented in Table 1 are not guaranteed to be tight, we used the code in Taylor, Hendrickx, and Glineur [31] (with SDP solvers [20, 28]) to compare tight (numerical) worst-case bounds for $N = 1, 2, 4, 10, 20, 30, 40, 47, 50$. These numerical worst-case bounds are guaranteed to be tight, i.e., equivalent to the bounds of either (P) or (P″), when the large-scale condition $d \geq N + 2$ is satisfied [31, Thm. 5], and we assume this condition hereafter. Tables 2 and 3 provide tight worst-case bounds for the decrease of the cost function $f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*)$ and the gradient norm decrease $\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)||$, respectively. Although most of the bounds in Table 1 are not guaranteed to be tight, the worst-case rate formulas in Table 1

Table 1

Asymptotic worst-case bounds on the cost function $\frac{1}{LR^2}(f(\boldsymbol{x}_N)-f(\boldsymbol{x}_*))$ *and the gradient norm* $\min_{i\in\{0,\ldots,N\}}\frac{1}{LR}||\nabla f(\boldsymbol{x}_i)||$ *of GM, FGM, OGM, OGM-m, OGM-OG, and OGM-a. (The worst-case cost function bound for OGM-m in the table corresponds to the bound for OGM after m iterations, because we do not have an analytical bound for the final iterate.*

Algorithm	Asymptotic worst-case bound		Require selecting
	Cost function	Gradient norm	N in advance
GM	$\frac{1}{4}N^{-1}$	$\sqrt{2}N^{-1}$	No
FGM	$2N^{-2}$	$2\sqrt{3}N^{-1.5}$	No
OGM	N^{-2}	$\sqrt{2}N^{-1}$	No
OGM-$m=\lfloor 2N/3\rfloor$	$\frac{9}{4}N^{-2}$	$\frac{3\sqrt{6}}{2}N^{-1.5}$	Yes
OGM-OG	$2N^{-2}$	$\sqrt{6}N^{-1.5}$	Yes
OGM-a ($a>2$)	$\frac{a}{2}N^{-2}$	$\frac{a\sqrt{6}}{2\sqrt{a-2}}N^{-1.5}$	No
OGM-$a=4$	$2N^{-2}$	$2\sqrt{3}N^{-1.5}$	

Table 2

Tight worst-case bounds on $\frac{LR^2}{f(\boldsymbol{x}_N)-f(\boldsymbol{x}_*)}$, *the reciprocal of the cost function, of GM, FGM, OGM, OGM-*$m=\lfloor 2N/3\rfloor$, *OGM-OG, and OGM-*$a=4$. *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 its simple modeling of bounds.*

N	GM	FGM	OGM	OGM-m	OGM-OG	OGM-a
1	6.0	6.0	8.0	6.0	7.3	6.5
2	10.0	11.1	16.2	12.0	13.2	15.1
4	18.0	24.7	39.1	24.2	28.6	32.3
10	42.0	90.7	159.1	86.6	99.9	106.4
20	82.0	283.6	525.1	275.3	310.4	308.9
30	122.0	578.6	1095.6	565.1	604.9	610.9
40	162.0	975.1	1869.2	899.0	1009.9	1012.8
47	190.0	1312.9	2531.1	1227.9	1352.8	1353.6
50	202.0	1472.8	**2845.1**	1374.4	1516.0	1514.6
Empi. $O(\cdot)$	$N^{-1.0}$	$N^{-1.9}$	$N^{-1.9}$	$N^{-1.8}$	$N^{-1.8}$	$N^{-1.8}$
Known $O(\cdot)$	N^{-1}	N^{-2}	N^{-2}	N^{-2}	N^{-2}	N^{-2}

are similar to the tight numerical results in Tables 2 and 3, except that the gradient bounds of OGM-m in Table 1 are relatively looser than those of OGM-m in Table 3. In particular, the tight numerical gradient bound of OGM-m is smaller than that of FGM in Table 3, which was not expected from their known (possibly loose) analytical bounds in Table 1.

6.3. Tight worst-case bounds on the gradient norm at the final iterate.

To be clear, section 5 and Tables 1 and 3 have focused on analyzing the *smallest* gradient norm among all iterates using the gradient form of PEP, whereas the gradient analysis in section 3.2 considers the *final* gradient in addition to the *smallest* gradient among all iterates. As mentioned before, we have not yet found a relaxation on the *final* gradient form of the PEP that provides as comparable results as for the relaxation on the *smallest* gradient form of PEP (P$''$) in section 5. To complete comparisons on the worst-case gradient bounds, Table 4 uses the code provided by Taylor, Hendrickx, and Glineur [31] (with SDP solvers [20, 28]) to compare tight (numerical) worst-case

Table 3

Tight worst-case bounds on $\frac{LR}{\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{x}_i)||}$*, the reciprocal of the gradient norm, of GM, FGM, OGM, OGM-*$m = \lfloor 2N/3 \rfloor$*, OGM-OG, and OGM-*$a = 4$*. Empirical rates were computed as described in Table* 2.

N	GM	FGM	OGM	OGM-m	OGM-OG	OGM-a
1	2.0	2.0	2.0	2.0	2.3	1.8
2	3.0	3.3	2.8	3.5	3.7	3.3
4	5.0	5.9	4.4	6.4	6.8	5.7
10	11.0	13.8	8.9	18.0	18.9	15.3
20	21.0	32.8	16.2	43.1	45.4	35.2
30	31.0	56.4	23.4	74.4	78.6	59.4
40	41.0	83.6	30.6	110.7	116.9	87.1
47	48.0	104.7	35.6	138.9	146.6	108.4
50	51.0	114.2	37.7	151.4	**160.0**	118.0
Empi. $O(\cdot)$	$N^{-1.0}$	$N^{-1.4}$	$N^{-0.9}$	$N^{-1.4}$	$N^{-1.4}$	$N^{-1.4}$
Known $O(\cdot)$	N^{-1}	$N^{-1.5}$	N^{-1}	$N^{-1.5}$	$N^{-1.5}$	$N^{-1.5}$

bounds on the *final* gradient of the FSFOMs presented in this paper.[5] The worst-case *smallest* gradient norm bounds (3.12) and (3.16) of GM and OGM-m, respectively (among algorithms considered), extend to the *final* gradient bounds.

In Table 4, FGM and OGM-$a = 4$ have slow $O(1/N)$ tight worst-case bounds on the final gradient, unlike OGM-$m = \lfloor 2N/3 \rfloor$ and OGM-OG, which roughly have $O(1/N^{1.5})$ bounds for both the smallest and final gradients. Theorem 3.5 has shown that the final gradient of OGM-m satisfies a worst-case rate $O(1/N^{1.5})$, but this is yet unknown for OGM-OG, which we leave for future work. We also leave for future work the challenge of developing an FSFOM that has $O(1/N^{1.5})$ or even faster worst-case rates for the final gradient decrease that are lower than those of OGM-m and OGM-OG, possibly without requiring to choose N in advance.

6.4. Nonoptimality of OGM-OG in terms of the worst-case gradient bound. Because OGM is optimal in terms of the function decrease when $d \geq N+1$ [7], one might hope that the OGM-OG would achieve the optimal worst-case bound in terms of the gradient decrease, since OGM-OG is also derived by optimizing the step coefficients over the gradient form of relaxed PEP. However, the OGM-OG is apparently not optimal, as explained next.

Taylor, Hendrickx, and Glineur [31] numerically studied an optimal fixed-step GM using their tight PEP in terms of both the cost function and gradient decrease. In other words, they searched for an optimal step h of GM,

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{h}{L}\nabla f(\boldsymbol{x}_i)$$

for $i = 0, \dots, N-1$, and a given N with respect to either $f(\boldsymbol{x}_N) - f(x_*)$ or $||f(\boldsymbol{x}_N)||$. In the special case of $N = 1$, Taylor, Hendrickx, and Glineur [31] numerically conjectured

[5] Table 4 reports tight worst-case gradient bounds for both the *final* primary iterate $\boldsymbol{y}_N$ and the *final* secondary iterate $\boldsymbol{x}_N$ (if necessary), unlike Tables 2 and 3. We observed that numerical tight worst-case cost function bounds on both final iterates $\boldsymbol{y}_N$ and $\boldsymbol{x}_N$ have similar values for the algorithms in Table 2 (unlike Table 4), so we did not report the bounds on $\boldsymbol{y}_N$ for simplicity. We also did not report numerical tight *smallest* worst-case gradient norm bounds $\min_{i\in\{0,\dots,N\}} ||\nabla f(\boldsymbol{y}_i)||$ of the primary iterates $\{\boldsymbol{y}_i\}$ because the code provided by Taylor, Hendrickx, and Glineur [31] does not support computing their values, unlike that of the secondary iterates $\{\boldsymbol{x}_i\}$ in Table 3.

TABLE 4

Tight worst-case bounds on $\frac{LR}{||\nabla f(\boldsymbol{x}_N)||}$ $\left(\textit{and } \frac{LR}{||\nabla f(\boldsymbol{y}_N)||}\right)$*, the reciprocal of the final gradient norm, of GM, FGM, OGM, OGM-m*$=\lfloor 2N/3 \rfloor$*, OGM-OG, and OGM-a*$=4$*. Empirical rates were computed as described in Table 2. The known bounds of OGM-OG and OGM-a* $= 4$ *are derived based on section 3.2, where the empirical bounds of OGM-OG are comparable to the bounds of OGM-m*$=\lfloor 2N/3 \rfloor$ *with known rate* $O(1/N^{1.5})$.

N	GM	FGM		OGM		OGM-m	OGM-OG	OGM-a	
		$\boldsymbol{y}_N$	$\boldsymbol{x}_N$	$\boldsymbol{y}_N$	$\boldsymbol{x}_N$			$\boldsymbol{y}_N$	$\boldsymbol{x}_N$
1	2.0	2.0	2.0	2.0	2.0	2.0	2.3	2.0	1.8
2	3.0	3.0	3.3	3.2	2.8	3.5	3.7	3.6	3.3
4	5.0	5.8	5.9	5.5	4.4	6.4	6.8	6.8	5.1
10	11.0	15.1	8.2	11.9	8.9	18.0	18.9	15.9	8.7
20	21.0	25.1	13.1	22.2	16.2	43.1	44.4	26.3	13.8
30	31.0	35.1	18.2	32.4	23.4	74.4	74.1	36.3	18.8
40	41.0	45.2	23.2	42.5	30.6	110.7	107.0	46.3	23.8
47	48.0	52.2	26.7	49.6	35.6	138.9	131.6	53.3	27.3
50	51.0	55.3	28.2	52.6	37.7	**151.4**	142.6	56.3	28.8
Empi. $O(\cdot)$	$N^{-1.0}$	$N^{-0.9}$		$N^{-0.9}$		$N^{-1.4}$	$N^{-1.3}$	$N^{-0.9}$	
Known $O(\cdot)$	N^{-1}	N^{-1}		N^{-1}		$N^{-1.5}$	N^{-1}	N^{-1}	

that the step size $h = 1.5$ is optimal in terms of the cost function decrease. The corresponding GM is equivalent to OGM for $N = 1$, and this (numerically) confirms the optimality of OGM [7] for $N = 1$. They also numerically conjectured that the optimal step size of GM for $N = 1$ in terms of the gradient decrease is $h = \sqrt{2}$ with a worst-case bound

$$||\nabla f(\boldsymbol{x}_1)|| \leq \frac{LR}{\sqrt{2}+1} \approx \frac{LR}{2.4}. \tag{6.1}$$

However, OGM-OG for $N = 1$ reduces to GM with $h = \frac{4}{3} \approx 1.3$ with a bound $\frac{LR}{2.3}$ in Tables 3 and 4, implying that OGM-OG is not optimal even for $N = 1$ based on the numerical evidence in [31].

This analysis for $N = 1$ illustrates that there is still room for improvement in accelerating the worst-case rate of first-order methods in terms of gradients, which we leave for future work, possibly with a tighter relaxation on the gradient form of PEP. In addition, we leave for future work studying the optimal worst-case bound for the gradient decrease of first-order methods building upon [7, 23], and developing an FSFOM that achieves such an optimal bound. Nevertheless, the OGM-OG is the best known FSFOM for decreasing the gradient norm among the class FSFOM and will be useful when decreasing the gradient is key.

7. Conclusion.

We generalized the formulation of OGM and analyzed its worst-case bounds on the function value and gradient, using the cost function form and the gradient form of relaxed PEP. We then proposed OGM-OG by optimizing the step coefficients of FSFOM using a relaxed PEP with respect to the gradient, similar to the development of the (optimal) OGM. To the best of our knowledge, the worst-case bound on the gradient of the OGM-OG is the best known analytical worst-case bound for decreasing the gradient norm among the class FSFOM.

However, this OGM-OG is not optimal for decreasing the gradient norm, and further accelerating the worst-case rate of FSFOM in terms of the gradient possibly with a tight relaxation on the gradient form of PEP is a possible research direction. On the other hand, deriving an optimal worst-case bound for the gradient norm of first-order methods, similar to that for the function decrease [7], will be useful. Nonetheless, the proposed OGM-OG (and OGM-a) may be useful when one finds minimizing gradients important, particularly in dual problems. In addition, we used the proposed gradient form of PEP to show that FGM decreases the (smallest) gradient with a rate $O(1/N^{1.5})$, implying that FGM is comparable in a big-O sense to OGM-m, OGM-OG, and OGM-a for the gradient decrease.

Our analysis considers unconstrained smooth convex minimization; extending such gradient norm worst-case analysis to constrained problems or nonsmooth composite convex problems is a natural direction to pursue, which is studied for FGM (or FISTA [2]) by the authors [17]. In addition, extending the analyses on the general form of FGM in [1, 3, 29] to GOGM is a possible research direction. Lastly, investigating a new relaxation of the PEP approach that allows adaptive step size such as backtracking line search or exact line search [5, 8] is of interest.

Software. https://gitlab.eecs.umich.edu/michigan-fast-optimization has MATLAB codes for the algorithms considered and the SDP approaches in sections 5.4 and 6.

Appendix A. Proof of Theorem 3.4. Due to (3.10), we have

$$\max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} \min_{i\in\{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \le \max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} ||\nabla f(\boldsymbol{x}_N)|| \le \frac{LR}{\theta_N},$$

and the rest of the proof shows that the following inequality holds:

$$\begin{aligned}\frac{LR}{\theta_N} &= \min_{i\in\{0,\ldots,N\}} ||\nabla\phi(\boldsymbol{x}_i)|| = ||\nabla\phi(\boldsymbol{x}_N)|| \\ &\le \max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} \min_{i\in\{0,\ldots,N\}} ||\nabla f(\boldsymbol{x}_i)|| \le \max_{\substack{f\in\mathcal{F}_L(\mathbb{R}^d),\\ \boldsymbol{x}_*\in X_*(f),\\ ||\boldsymbol{x}_0-\boldsymbol{x}_*||\le R}} ||\nabla f(\boldsymbol{x}_N)||,\end{aligned} \tag{A.1}$$

which then implies (3.15) with $\theta_N \ge \frac{N+1}{\sqrt{2}}$ (3.6).

Starting from $\boldsymbol{x}_0 = R\boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a unit vector, we first use induction to show that the iterates

$$\boldsymbol{x}_i = (-1)^i\frac{1}{\theta_i}R\boldsymbol{\nu}, \quad i = 0,\ldots,N, \tag{A.2}$$

correspond to the iterates of OGM applied to $\phi(\boldsymbol{x})$. We use [15, Prop. 4] that the sequence generated by OGM is identical to the sequence generated by FSFOM with

$$h_{i+1,k} = \begin{cases} \frac{\theta_i-1}{\theta_{i+1}}h_{i,k}, & k = 0,\ldots,i-2,\\ \frac{\theta_i-1}{\theta_{i+1}}(h_{i,i-1}-1), & k = i-1,\\ 1+\frac{2\theta_i-1}{\theta_{i+1}}, & k = i,\end{cases} \tag{A.3}$$

for $i = 0,\ldots,N-1$.

Assuming that (A.2) holds for $i < N$, we have

$$\begin{aligned}\boldsymbol{x}_{i+1} &= \boldsymbol{x}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k}\nabla\phi(\boldsymbol{x}_k)\\ &= \boldsymbol{x}_i - \frac{1}{L}\left(1+\frac{2\theta_i-1}{\theta_{i+1}}\right)\nabla\phi(\boldsymbol{x}_i) - \frac{1}{L}\sum_{k=0}^{i-1}\frac{\theta_i-1}{\theta_{i+1}}h_{i,k}\nabla\phi(\boldsymbol{x}_k) + \frac{1}{L}\frac{\theta_i-1}{\theta_{i+1}}\nabla\phi(\boldsymbol{x}_{i-1})\end{aligned}$$

$$= \frac{1-2\theta_i}{\theta_{i+1}}\boldsymbol{x}_i + \frac{\theta_i - 1}{\theta_{i+1}}(\boldsymbol{x}_i - \boldsymbol{x}_{i-1}) + \frac{\theta_i - 1}{\theta_{i+1}}\boldsymbol{x}_{i-1}$$
$$= -\frac{\theta_i}{\theta_{i+1}}\boldsymbol{x}_i = (-1)^{i+1}\frac{1}{\theta_{i+1}}R\boldsymbol{\nu},$$

using (A.3) and $\nabla\phi(\boldsymbol{x}) = L\boldsymbol{x}$. Therefore, after N iterations of OGM we have

$$\text{(A.4)} \qquad \min_{i\in\{0,\dots,N\}} ||\nabla\phi(\boldsymbol{x}_i)|| = ||\nabla\phi(\boldsymbol{x}_N)|| = \left|\left|\nabla\phi\left((-1)^N\frac{1}{\theta_N}R\boldsymbol{\nu}\right)\right|\right| = \frac{LR}{\theta_N},$$

which is equivalent to (A.1). The first equality of (A.4) holds since OGM monotonically decreases the gradient norm of $\phi(\boldsymbol{x})$, i.e., $||\nabla\phi(\boldsymbol{x}_i)|| = \frac{LR}{\theta_i}$ monotonically decreases as i increases. □

Appendix B. Proof of Lemma 4.1. It is obvious that $(\boldsymbol{\lambda}, \boldsymbol{\tau})$ in (4.9) is in Λ (4.5). Using (4.2) and (4.4), we have

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

and inserting (4.8) and (4.9) yields

$$\begin{aligned} &\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) \\ &= \begin{cases} \frac{1}{2}\left(\Omega_i\tau_0\frac{\theta_i}{\Omega_i}\left(2\theta_k - \sum_{j=k+1}^{i-1} h_{j,k}\right) + \theta_i\tau_0\sum_{j=k+1}^{i-1} h_{j,k}\right), & i = 2,\dots,N-1, \\ & \quad k = 0,\dots,i-2, \\ \frac{1}{2}\left(\frac{\Omega_N}{2}\tau_0\frac{\theta_N}{\Omega_N}\left(2\theta_k - \sum_{j=k+1}^{N-1} h_{j,k}\right) + \frac{\theta_N}{2}\tau_0\sum_{j=k+1}^{N-1} h_{j,k}\right), & i = N,\ k = 0,\dots,i-2, \\ \frac{1}{2}\left(\Omega_i\tau_0\left(1 + \frac{(2\theta_{i-1}-1)\theta_i}{\Omega_i}\right) - \Omega_{i-1}\tau_0\right), & i = 1,\dots,N-1, \\ & \quad k = i-1, \\ \frac{1}{2}\left(\frac{\Omega_N}{2}\tau_0\left(1 + \frac{(2\theta_{N-1}-1)\theta_N}{\Omega_N}\right) - \Omega_{N-1}\tau_0\right), & i = N,\ k = i-1, \\ \Omega_i\tau_0, & i = 0,\dots,N-1,\ k = i, \\ \frac{\Omega_N}{4}\tau_0, & i = N,\ k = i, \end{cases} \\ &= \begin{cases} \theta_i\theta_k\tau_0, & i = 1,\dots,N-1,\ k = 0,\dots,i-1, \\ \frac{\theta_N\theta_k}{2}\tau_0, & i = N,\ k = 0,\dots,i-1, \\ \Omega_i\tau_0, & i = 1,\dots,N-1,\ k = i, \\ \frac{\Omega_N}{4}\tau_0, & i = N,\ k = i, \end{cases} \end{aligned}$$

for θ_i and Ω_i in (4.10). Then, using (4.9) and (4.10), we show the feasibility condition of (D):

$$\begin{pmatrix} \boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} = \left(\mathsf{diag}\{\boldsymbol{\Omega} - \boldsymbol{\theta}^2\} + \boldsymbol{\theta}\boldsymbol{\theta}^\top\right)\tau_0 \succeq \boldsymbol{0},$$

where $\boldsymbol{\theta} = \left(\theta_0, \ldots, \theta_{N-1}, \frac{\theta_N}{2}, \frac{1}{2}\right)^\top$ and $\boldsymbol{\Omega} = \left(\Omega_0, \ldots, \Omega_{N-1}, \frac{\Omega_N}{4}, \frac{1}{4}\right)^\top$. □

Appendix C. Proof of Lemma 4.2. It is obvious that $(\boldsymbol{\lambda}, \boldsymbol{\tau})$ in (4.12) is in Λ (4.5). Inserting (4.11) and (4.12) into (B.1) yields

$$\begin{aligned}
&\boldsymbol{S}(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) + \frac{1}{2}\boldsymbol{u}_N\boldsymbol{u}_N^\top \\
&= \begin{cases} \frac{1}{2}\left(T_i\tau_0\frac{t_i}{T_i}\left(2t_k - \sum_{j=k+1}^{i-1} h_{j,k}\right) + t_i\tau_0\sum_{j=k+1}^{i-1} h_{j,k}\right), & i = 2, \ldots, N-1, \\ & \quad k = 0, \ldots, i-2, \\ \frac{1}{2}\left(T_i\tau_0\left(1 + \frac{(2t_{i-1}-1)t_i}{T_i}\right) - T_{i-1}\tau_0\right), & i = 1, \ldots, N,\ k = i-1, \\ T_i\tau_0, & i = 0, \ldots, N,\ k = i \end{cases} \\
&= \begin{cases} t_it_k\tau_0, & i = 1, \ldots, N,\ k = 0, \ldots, i-1, \\ T_i\tau_0, & i = 1, \ldots, N,\ k = i, \end{cases}
\end{aligned}$$

for t_i and T_i in (4.13). Then, using (4.12) and (4.13), we show the feasibility condition of (D′):

$$\begin{pmatrix} \boldsymbol{S}'(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} = \left(\mathsf{diag}\left\{\boldsymbol{T} - \boldsymbol{t}^2\right\} + \boldsymbol{t}\boldsymbol{t}^\top\right)\tau_0 \succeq \boldsymbol{0},$$

where $\boldsymbol{t} = \left(t_0, \ldots, t_N, \frac{1}{2}\right)^\top$ and $\boldsymbol{T} = \left(T_0, \ldots, T_N, \frac{1}{4}\right)^\top$. □

Appendix D. Proof of Proposition 4.3. The proof consists of three propositions, and they follow the derivations in [15, Props. 3, 4, and 5], respectively. Note that this proof is independent of the choice of θ_i and Ω_i.

Proposition D.1. *The step coefficient* (4.8) *satisfies the following recursive relationship:*

$$h_{i+1,k} = \begin{cases} \frac{(\Omega_i - \theta_i)\theta_{i+1}}{\theta_i\Omega_{i+1}}h_{i,k} & k = 0, \ldots, i-2, \\ \frac{(\Omega_i - \theta_i)\theta_{i+1}}{\theta_i\Omega_{i+1}}(h_{i,i-1} - 1), & k = i-1, \\ 1 + \frac{(2\theta_i - 1)\theta_{i+1}}{\Omega_{i+1}}, & k = i, \end{cases} \tag{D.1}$$

for $i = 0, \ldots, N-1$.

Proof. We use the notation $h'_{i,k}$ for the coefficients (4.8) to distinguish them from (D.1). It is obvious that $h'_{i+1,i} = h_{i+1,i}, i = 0, \ldots, N-1$, and we clearly have

$$\begin{aligned}
h'_{i+1,i-1} &= \frac{\theta_{i+1}}{\Omega_{i+1}}\left(2\theta_{i-1} - h'_{i,i-1}\right) = \frac{\theta_{i+1}}{\Omega_{i+1}}\left(2\theta_{i-1} - \left(1 + \frac{(2\theta_{i-1} - 1)\theta_i}{\Omega_i}\right)\right) \\
&= \frac{(2\theta_{i-1} - 1)(\Omega_i - \theta_i)\theta_{i+1}}{\Omega_i\Omega_{i+1}} = \frac{(\Omega_i - \theta_i)\theta_{i+1}}{\theta_i\Omega_{i+1}}(h_{i,i-1} - 1) = h_{i+1,i-1}.
\end{aligned}$$

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

$$\begin{aligned}
h'_{n+1,k} &= \frac{\theta_{n+1}}{\Omega_{n+1}}\left(2\theta_k - \sum_{j=k+1}^{n} h'_{j,k}\right) = \frac{\theta_{n+1}}{\Omega_{n+1}}\left(2\theta_k - \sum_{j=k+1}^{n-1} h'_{j,k} - h'_{n,k}\right) \\
&= \frac{\theta_{n+1}}{\Omega_{n+1}}\left(\frac{\Omega_n}{\theta_n}h'_{n,k} - h'_{n,k}\right) = \frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}h_{n,k} = h_{n+1,k}.
\end{aligned}$$

□

PROPOSITION D.2. *The sequence $\{\boldsymbol{x}_0, \dots, \boldsymbol{x}_N\}$ generated by FSFOM with* (D.1) *is identical to the corresponding sequence generated by GOGM*1*.*

Proof. We use induction, and for clarity we use the notation $\boldsymbol{x}_0', \dots, \boldsymbol{x}_N'$ for FSFOM with (D.1). It is obvious that $\boldsymbol{x}_0' = \boldsymbol{x}_0$, and we have

$$\begin{aligned}\boldsymbol{x}_1' &= \boldsymbol{x}_0' - \frac{1}{L}h_{1,0}\nabla f(\boldsymbol{x}_0') = \boldsymbol{x}_0 - \frac{1}{L}\left(1 + \frac{(2\theta_0 - 1)\theta_1}{\Omega_1}\right)\nabla f(\boldsymbol{x}_0)\\ &= \boldsymbol{y}_1 + \frac{(\Omega_0 - \theta_0)\theta_1}{\theta_0\Omega_1}(\boldsymbol{y}_1 - \boldsymbol{y}_0) + \frac{(2\theta_0^2 - \Omega_0)\theta_1}{\theta_0\Omega_1}(\boldsymbol{y}_1 - \boldsymbol{x}_0) = \boldsymbol{x}_1.\end{aligned}$$

Assuming $\boldsymbol{x}_i' = \boldsymbol{x}_i$ for $i = 0, \dots, n$, we then have

$$\begin{aligned}\boldsymbol{x}_{n+1}' &= \boldsymbol{x}_n' - \frac{1}{L}h_{n+1,n}\nabla f(\boldsymbol{x}_n') - \frac{1}{L}h_{n+1,n-1}\nabla f(\boldsymbol{x}_{n-1}') - \frac{1}{L}\sum_{k=0}^{n-2}h_{n+1,k}\nabla f(\boldsymbol{x}_k')\\
&= \boldsymbol{x}_n - \frac{1}{L}\left(1 + \frac{(2\theta_n - 1)\theta_{n+1}}{\Omega_{n+1}}\right)\nabla f(\boldsymbol{x}_n) - \frac{1}{L}\frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}(h_{n,n-1} - 1)\nabla f(\boldsymbol{x}_{n-1})\\
&\quad - \frac{1}{L}\sum_{k=0}^{n-2}\frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}h_{n,k}\nabla f(\boldsymbol{x}_k)\\
&= \boldsymbol{y}_{n+1} - \frac{1}{L}\frac{(2\theta_n^2 - \Omega_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}\nabla f(\boldsymbol{x}_n)\\
&\quad - \frac{1}{L}\frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}\left(\nabla f(\boldsymbol{x}_n) - \nabla f(\boldsymbol{x}_{n-1}) + \sum_{k=0}^{n-1}h_{n,k}\nabla f(\boldsymbol{x}_k)\right)\\
&= \boldsymbol{y}_{n+1} + \frac{(2\theta_n^2 - \Omega_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}(\boldsymbol{y}_{n+1} - \boldsymbol{x}_n)\\
&\quad + \frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}\left(-\frac{1}{L}\nabla f(\boldsymbol{x}_n) + \frac{1}{L}\nabla f(\boldsymbol{x}_{n-1}) + \boldsymbol{x}_n - \boldsymbol{x}_{n-1}\right)\\
&= \boldsymbol{y}_{n+1} + \frac{(\Omega_n - \theta_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}(\boldsymbol{y}_{n+1} - \boldsymbol{y}_n) + \frac{(2\theta_n^2 - \Omega_n)\theta_{n+1}}{\theta_n\Omega_{n+1}}(\boldsymbol{y}_{n+1} - \boldsymbol{x}_n) = \boldsymbol{x}_{n+1}.\end{aligned}$$ ◻

PROPOSITION D.3. *The sequence $\{\boldsymbol{x}_0, \dots, \boldsymbol{x}_N\}$ generated by FSFOM with* (4.8) *is identical to the corresponding sequence generated by GOGM*2*.*

Proof. We use induction, and for clarity we use the notation $\boldsymbol{x}_0', \dots, \boldsymbol{x}_N'$ for FSFOM with (4.8). It is obvious that $\boldsymbol{x}_0' = \boldsymbol{x}_0$, and we have

$$\begin{aligned}\boldsymbol{x}_1' &= \boldsymbol{x}_0' - \frac{1}{L}h_{1,0}\nabla f(\boldsymbol{x}_0') = \boldsymbol{x}_0 - \frac{1}{L}\left(1 + \frac{(2\theta_0 - 1)\theta_1}{\Omega_1}\right)\nabla f(\boldsymbol{x}_0)\\
&= \left(1 - \frac{\theta_1}{\Omega_1}\right)\left(\boldsymbol{x}_0 - \frac{1}{L}\nabla f(\boldsymbol{x}_0)\right) + \frac{\theta_1}{\Omega_1}\left(\boldsymbol{x}_0 - \frac{1}{L}\nabla f(\boldsymbol{x}_0) - \frac{1}{L}(2\theta_0 - 1)\nabla f(\boldsymbol{x}_0)\right)\\
&= \left(1 - \frac{\theta_1}{\Omega_1}\right)\boldsymbol{y}_1 + \frac{\theta_1}{\Omega_1}\boldsymbol{z}_1 = \boldsymbol{x}_1.\end{aligned}$$

Assuming $\boldsymbol{x}_i' = \boldsymbol{x}_i$ for $i = 0, \dots, n$, we then have

$$\begin{aligned}
&\boldsymbol{x}_{n+1}' = \boldsymbol{x}_n' - \frac{1}{L} h_{n+1,n} \nabla f(\boldsymbol{x}_n') - \frac{1}{L} \sum_{k=0}^{n-1} h_{n+1,k} \nabla f(\boldsymbol{x}_k') \\
&= \boldsymbol{x}_n - \frac{1}{L}\left(1 + \frac{(2\theta_n - 1)\theta_{n+1}}{\Omega_{n+1}}\right) \nabla f(\boldsymbol{x}_n) - \frac{1}{L} \sum_{k=0}^{n-1} \frac{\theta_{n+1}}{\Omega_{n+1}} \left(2\theta_k - \sum_{j=k+1}^{n} h_{j,k}\right) \nabla f(\boldsymbol{x}_k) \\
&= \left(1 - \frac{\theta_{n+1}}{\Omega_{n+1}}\right)\left(\boldsymbol{x}_n - \frac{1}{L}\nabla f(\boldsymbol{x}_n)\right) \\
&\qquad + \frac{\theta_{n+1}}{\Omega_{n+1}} \left(\boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} 2\theta_k \nabla f(\boldsymbol{x}_k) + \frac{1}{L} \sum_{k=0}^{n-1} \sum_{j=k+1}^{n} h_{j,k} \nabla f(\boldsymbol{x}_k)\right) \\
&= \left(1 - \frac{\theta_{n+1}}{\Omega_{n+1}}\right)\left(\boldsymbol{x}_n - \frac{1}{L}\nabla f(\boldsymbol{x}_n)\right) + \frac{\theta_{n+1}}{\Omega_{n+1}} \left(\boldsymbol{x}_0 - \frac{1}{L} \sum_{k=0}^{n} 2\theta_k \nabla f(\boldsymbol{x}_k)\right) \\
&= \left(1 - \frac{\theta_{n+1}}{\Omega_{n+1}}\right) \boldsymbol{y}_{n+1} + \frac{\theta_{n+1}}{\Omega_{n+1}} \boldsymbol{z}_{n+1}. \qquad \square
\end{aligned}$$

Appendix E. Proof of Lemma 5.1. It is obvious that $(\boldsymbol{\lambda}, \eta, \boldsymbol{\tau}, \boldsymbol{\beta})$ in (5.5) and (5.6) is in Λ'' (5.3). Using (5.2) and (B.1), we have

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

and inserting (5.4), (5.5), and (5.6) yields

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

for t_i in (3.3). Then, using (5.5), we finally show that the feasibility condition of (D$''$) holds:

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

where $\boldsymbol{t} = (t_0, \dots, t_N, 1)^\top$. $\square$

Appendix F. Proof of Lemma 5.3. It is obvious that $(\boldsymbol{\lambda}, \eta, \boldsymbol{\tau}, \boldsymbol{\beta})$ in (5.9) and (5.10) is in Λ''. Inserting (4.11), (5.9), and (5.10) into (E.1) yields

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

for t_i and T_i in (4.13). Then, using (5.9), we finally show that the feasibility condition of (D$''$) holds:

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

where $\boldsymbol{t} = \left(t_0, \dots, t_N, \frac{1}{2}\right)^\top$. □

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

REFERENCES

[1] H. Attouch, Z. Chbani, J. Peypouquet, and P. Redont, *Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity*, Math. Program., 168 (2018), pp. 123–175, https://doi.org/10.1007/s10107-016-0992-8.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[3] A. Chambolle and C. Dossal, *On the convergence of the iterates of the "Fast iterative shrinkage/thresholding algorithm,"* J. Optim. Theory Appl., 166 (2015), pp. 968–982, https://doi.org/10.1007/s10957-015-0746-4.

[4] CVX Research Inc., *CVX: MATLAB Software for Disciplined Convex Programming, Version* 2.0, http://cvxr.com/cvx, 2012.

[5] E. de Klerk, F. Glineur, and A. B. Taylor, *On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions*, Optim. Lett., (2017), https://doi.org/10.1007/s11590-016-1087-4.

[6] O. Devolder, F. Glineur, and Y. Nesterov, *Double smoothing technique for large-scale linearly constrained convex optimization*, SIAM J. Optim., 22 (2012), pp. 702–727, https://doi.org/10.1137/110826102.

[7] Y. Drori, *The exact information-based complexity of smooth convex minimization*, J. Complexity, 39 (2017), pp. 1–16, https://doi.org/10.1016/j.jco.2016.11.001.

[8] Y. Drori and A. B. Taylor, *Efficient First-Order Methods for Convex Minimization: A Constructive Approach*, preprint, http://arxiv.org/abs/1803.05676, 2018.

[9] Y. Drori and M. Teboulle, *Performance of first-order methods for smooth convex minimization: A novel approach*, Math. Program., 145 (2014), pp. 451–482, https://doi.org/10.1007/s10107-013-0653-0.

[10] Y. Drori and M. Teboulle, *An optimal variant of Kelley's cutting-plane method*, Math. Program., 160 (2016), pp. 321–351, https://doi.org/10.1007/s10107-016-0985-7.

[11] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99, https://doi.org/10.1007/s10107-015-0871-8.

[12] M. Grant and S. Boyd, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control, V. Blondel, S. Boyd, and H. Kimura, eds., Lect. Notes Control Inform. Sci. 371, Springer-Verlag Limited, 2008, pp. 95–110; http://stanford.edu/~boyd/papers/graph_dcp.html.

[13] O. Güler, *New proximal point algorithms for convex minimization*, SIAM J. Optim., 2 (1992), pp. 649–664, https://doi.org/10.1137/0802032.

[14] D. Kim and J. A. Fessler, *An optimized first-order method for image restoration*, in Proc. IEEE Intl. Conf. on Image Processing, 2015, pp. 3675–3679, https://doi.org/10.1109/ICIP.2015.7351490.

[15] D. Kim and J. A. Fessler, *Optimized first-order methods for smooth convex minimization*, Math. Program., 159 (2016), pp. 81–107, https://doi.org/10.1007/s10107-015-0949-3.

[16] D. Kim and J. A. Fessler, *On the convergence analysis of the optimized gradient methods*, J. Optim. Theory Appl., 172 (2017), pp. 187–205, https://doi.org/10.1007/s10957-016-1018-7.

[17] D. Kim and J. A. Fessler, *Another look at the fast iterative shrinkage/thresholding algorithm (FISTA)*, SIAM J. Optim., 28 (2018), pp. 223–250, https://doi.org/10.1137/16M108940X.

[18] D. Kim and J. A. Fessler, *Optimizing the Efficiency of First-Order Methods for Decreasing the Gradient of Smooth Convex Functions*, preprint, http://arxiv.org/abs/1803.06600, 2018.

[19] L. Lessard, B. Recht, and A. Packard, *Analysis and design of optimization algorithms via integral quadratic constraints*, SIAM J. Optim., 26 (2016), pp. 57–95, https://doi.org/10.1137/15M1009597.

[20] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[21] R. D. C. Monteiro and B. F. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM J. Optim., 23 (2013), pp. 1092–1125, https://doi.org/10.1137/110833786.

[22] I. Necoara and A. Patrascu, *Iteration complexity analysis of dual first order methods for conic convex programming*, Optim. Methods Softw., 31 (2016), pp. 645–678, https://doi.org/10.1080/10556788.2016.1161763.

[23] A. S. Nemirovsky, *Information-based complexity of linear operator equations*, J. Complexity, 8 (1992), pp. 153–175, https://doi.org/10.1016/0885-064X(92)90013-2.

[24] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Dokl. Akad. Nauk. USSR, 269 (1983), pp. 543–547.

[25] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, 2004, https://doi.org/10.1007/978-1-4419-8853-9.

[26] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152, https://doi.org/10.1007/s10107-004-0552-5.

[27] Y. Nesterov, *How to make the gradients small*, Optima, 88 (2012), pp. 10–11, http://www.mathopt.org/?nav=optima_newsletter.

[28] J. Sturm, *Using SeDuMi* 1.02*, A MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11 (1999), pp. 625–653, https://doi.org/10.1080/10556789908805766.

[29] W. Su, S. Boyd, and E. J. Candès, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, J. Mach. Learning Res., 17 (2016), pp. 1–43, http://jmlr.org/papers/v17/15-084.html.

[30] A. B. Taylor, J. M. Hendrickx, and F. Glineur, *Exact worst-case performance of first-order methods for composite convex optimization*, SIAM J. Optim., 27 (2017), pp. 1283–1313, https://doi.org/10.1137/16m108104x.

[31] A. B. Taylor, J. M. Hendrickx, and F. Glineur, *Smooth strongly convex interpolation and exact worst-case performance of first- order methods*, Math. Program., 161 (2017), pp. 307–345, https://doi.org/10.1007/s10107-016-1009-3.