Adaptive Accelerated Composite Minimization

Reza Rahimi Baghbadorani, Sergio Grammatico, and Peyman Mohajerin Esfahani

Abstract. The choice of the stepsize in first-order convex optimization is typically based on the smoothness constant and plays a crucial role in the performance of algorithms. Recently, there has been a resurgent interest in introducing adaptive stepsizes that do not explicitly depend on smooth constant. In this paper, we propose a novel adaptive stepsize rule based on function evaluations (i.e., zero-order information) that enjoys provable convergence guarantees for both accelerated and non-accelerated gradient descent. We further discuss the similarities and differences between the proposed stepsize regimes and the existing stepsize rules (including Polyak and Armijo). Numerically, we benchmark the performance of our proposed algorithms with the state-of-the-art literature in three different classes of smooth minimization (logistic regression, quadratic programming, log-sum-exponential, and approximate semidefinite programming), composite minimization (ℓ_1 constrained and regularized problems), and non-convex minimization (cubic problem).

Keywords: Adaptive stepsize, first-order methods, composite convex optimization

1. Introduction

Thanks to its simple implementation and applicability, gradient descent (GD) is perhaps the most popular algorithm in convex optimization [5, 14]. The main challenge in using gradient descent is the choice of the right stepsize, which has a considerable impact on the convergence speed. Several works have studied this choice, yet either the fastest possible convergence rate cannot be guaranteed or an expensive linesearch at each iteration is needed [37, 24, 34, 15]. In this paper, we address these issues by introducing a novel and simple linesearch method.

Let us consider the unconstrained convex minimization problem $\min_{x\in\mathbb{R}^n} f(x)$, where f is a smooth convex function. The usual iterative algorithm is of the form

$$x_{k+1} = x_k + \lambda_k d_k. \tag{1}$$

where λ_k and d_k represent the stepsize and the descent direction, respectively. In the last decades, many algorithms have been developed for choosing $(\lambda_k)_{k\in\mathbb{N}}$ and $(d_k)_{k\in\mathbb{N}}$ in (1) either statically or adaptively. We review some classes of design choices in the following.

A key object playing an important role in determining most of the existing stepsize rules at the iteration x_k is the objective function along the update direction $d_k = -\nabla f(x_k)$ defined by

$$\phi_k(\lambda) := f(x_k + \lambda d_k). \tag{2}$$

Date: May 7, 2024.

The authors are with the Delft Center for Systems and Control, TU Delft, The Netherlands. (e-mails: R.Rahimibaghbadorani@tudelft.nl; S.Grammatico@tudelft.nl; P.MohajerinEsfahani@tudelft.nl.) This work was supported by the ERC under the research project TRUST-91562 and COSMOS-91409.

When the function f is convex, then the scaler function $\phi_k : \mathbb{R} \to \mathbb{R}$ defined in (2) is also convex. Several classical stepsize rules defined via ϕ_k in (2) are the following.

(i) GD with constant stepsize [34]**:** The simplest stepsize rule is the constant $\lambda_k = 1/2L$, where L is the so-called smoothness parameter (Lipschitz continuity of $\nabla f(x)$. The convergence rate of the suboptimality $f(x_k) - f(x^*)$ with the constant stepsize is $\mathcal{O}(k^{-1})$.

(ii) GD with exact linesearch [34]**:** Another classical stepsize is to find the optimal stepsize minimizing the function ϕ_k along the direction d_k is $\lambda_k = \arg\min_\lambda \phi_k(\lambda)$ which involves a scalar optimization problem that requires evaluation of the original function f.

(iii) GD with approximated stepsize [34]**:** Due to the complexity of finding the stepsize by solving the exact linesearch minimization, alternative stepsize rules have been proposed that only require ensuring an inequality such as $\phi_k(\lambda) \le \phi_k(0) + c\phi_k'(0)$ for some predefined constant $c > 0$. Among others, Armijo [2], Wolf, and strong Wolf linesearch [34] are backtracking linesearch methods falling into this category.

$$\text{Armijo}: \qquad \phi_k(\lambda) \le \phi_k(0) + c_1\lambda\phi_k'(0), \; c_1 \in (0,1). \tag{3a}$$

$$\text{Wolf}: \qquad \phi_k'(\lambda) \ge c_2\phi_k'(0), \; c_2 \in (c_1,1) + \text{Armijo condition}. \tag{3b}$$

$$\text{strong Wolf}: \qquad |\phi_k'(\lambda)| \le c_2\phi_k'(0), \; c_2 \in (c_1,1) + \text{Armijo condition}. \tag{3c}$$

The suboptimality convergence rate of these algorithms is $\mathcal{O}(k^{-1})$. There are also several stepsize rules approximating the smooth constant locally based on the zero-order oracle's information (i.e., the objective function evaluation) [36, 24, 3, 4]. In the following, we review two important stepsize rules that fall into this category:

(iv) GD with the Polyak stepsize [36]**:** One of the classic adaptive stepsize rules is the Polyak stepsize, which uses the zero-order oracle's information at each iteration to locally approximate the smooth constant of the objective function:

$$\lambda_k = \frac{f(x_k) - f(x^*)}{\|\nabla f(x_k)\|^2}. \tag{4}$$

Although the Polyak stepsize offers optimal convergence rates for GD (1) when the update direction is set to $d_k = -\nabla f(x_k)$ [16], it cannot be applicable in problems where the optimal function value $f(x^*)$ is not available. We will get back to the Polyak stepsize and its relation to our proposed rule in Subsection 2.3 (see equations (18)).

(v) GD with adaptive stepsize [24]**:** Recently, the next adaptive stepsize has been proposed based on the idea of approximating the smoothness constant locally:

$$\lambda_k = \min\left\{\sqrt{1+\theta_{k-1}}\lambda_{k-1}, \frac{\|x_k - x_{k-1}\|}{2\|\nabla f(x_k) - \nabla f(x_{k-1})\|}\right\}, \quad \theta_k = \frac{\lambda_k}{\lambda_{k-1}}. \tag{5}$$

The second term on the left-hand side of (5) represents this local approximation as the inverse of the smoothness constant, while the first term ensures a required convergence rate of this parameter. The theoretical sub-optimality convergence rate for the stepsize rule (5) is $\mathcal{O}(k^{-1})$, however, the adaptive (i.e., state-dependent) nature of (5) has the advantage of (a) lifting the knowledge of the smoothness constant for the objective function, and (b) improving the practical convergence rate when the local smoothness is smaller than the global constant one. Next, we consider methods with a dynamic direction d_k in (1). The update rules in this class of methods have an extra term, which is called momentum, and allows the descent direction to have some inertia in the search space [37].

(vi) AGD with constant stepsize [29]**:**

$$\begin{cases} \beta_k = \dfrac{1+\sqrt{1+4\beta_{k-1}^2}}{2}, \ \gamma_k = \dfrac{1-\beta_k}{\beta_{k+1}}, \\ \lambda_k = \dfrac{1}{L}, \ d_k = -\gamma_k \left(\dfrac{\lambda_{k-1}}{\lambda_k} d_{k-1} - (1+\dfrac{1}{\gamma_k}) \nabla f(x_k) + \dfrac{\lambda_{k-1}}{\lambda_k} \nabla f(x_{k-1}) \right), \end{cases}$$

where L is the smooth constant of f. The descent direction in this method uses previous iteration values to accelerate the convergence rate, which is $\mathcal{O}(k^{-2})$ for $f(x_k) - f(x^*)$.

(vii) AGD with adaptive stepsize [24]**:** A heuristic version of adaptive stepsize (5) for accelerated methods is also proposed in [24, Section 3]

$$\begin{cases} \lambda_k = \min\left\{ \sqrt{1+\dfrac{\theta_{k-1}}{2}}\lambda_{k-1}, \dfrac{\|x_k - x_{k-1}\|}{2\|\nabla f(x_k) - \nabla f(x_{k-1})\|} \right\}, \ \theta_k = \dfrac{\lambda_k}{\lambda_{k-1}}, \\ \Lambda_k = \min\left\{ \sqrt{1+\dfrac{\Theta_{k-1}}{2}}\Lambda_{k-1}, \dfrac{\|\nabla f(x_k) - \nabla f(x_{k-1})\|}{2\|x_k - x_{k-1}\|} \right\}, \ \Theta_k = \dfrac{\Lambda_k}{\Lambda_{k-1}}, \\ \beta_k = \dfrac{\sqrt{1/\lambda_k} - \sqrt{\Lambda_k}}{\sqrt{1/\lambda_k} + \sqrt{\Lambda_k}}, \\ d_k = \beta_k \left(\dfrac{\lambda_{k-1}}{\lambda_k} d_{k-1} - (1+\dfrac{1}{\beta_k}) \nabla f(x_k) + \dfrac{\lambda_{k-1}}{\lambda_k} \nabla f(x_{k-1}) \right). \end{cases}$$

This adaptive rule presents promising numerical performance while there is no formal theoretical guarantee explaining this interesting performance. This theoretical gap toward accelerated techniques is one of the motivations of this study.

We also consider the composite minimization problem

$$\min_{x\in\mathcal{X}} F(x) = \min_{x\in\mathcal{X}} f(x) + h(x) \tag{6}$$

where $f\colon \mathcal{X} \to \mathbb{R}$ is a convex and smooth function, and $h\colon \mathcal{X} \to \mathbb{R}$ is convex and possibly nonsmooth but prox-friendly (technical details are postponed to Section 2). There are several methods for convex composite minimization in the existing literature. Among them, subgradient descent and mirror descent are two well-known classic methods that suffer from slow convergence rate $\mathcal{O}(k^{-1/2})$ [28, 7, 42, 40]. Following the seminal works by Nesterov, one can exploit the structure of the nonsmooth part in the objective function and deploy a first-order accelerated method. ISTA and FISTA [8], which are extensions of GD and NAGD, respectively, offer a faster convergence rate compared to subgradient and mirror descent. However, these methods require knowledge of the smooth constant associated with the smooth part, f in (6). Additionally, there are various backtracking techniques available to approximate the smooth constant, but they often tend to be conservative, resulting in larger values than the actual smooth constant. This conservatism can impact the convergence speed. Recently, the authors in [39, 23] propose an algorithm that enjoys the local smoothness of f to determine the stepsize. However, despite its simplicity and efficacy, the convergence guarantee is still the suboptimal rate of $\mathcal{O}(k^{-1})$. Inspired by this, we develop an algorithm for convex composite minimization problems (6) with the best theoretical convergence rate $\mathcal{O}(k^{-2})$ while maintaining a reasonable level of computational complexity.

Contributions. Following the same spirit of the adaptive stepsize rules, this study proposes a novel stepsize rule without the knowledge of the global smoothness constant for solving convex composite minimization problems. Using a Lyapunov-based argument, we show that the proposed rule enjoys the optimal worst-case complexity bound of $\mathcal{O}(k^{-1})$ and $\mathcal{O}(k^{-2})$ in both cases of nonaccelerated and accelerated algorithms, respectively. These results are developed for a general class of composite convex functions and are optimal in the sense that they match the theoretical lower bounds of the class of first-order algorithms for this class of functions. Our convergence results are summarized in Table 1 where our update direction $d_k = -G^f_{\lambda h}(x_k)$ is our gradient mapping (see the definition (10) for more details concerning the gradient mapping). Let us note that the results in Table 1 are reported for the general composite case (6). If the nonsmooth term is $h(x) = 0$, then the gradient mapping reduces to $G^f_{\lambda h}(x) = \nabla f(x)$. In the rest of the introduction, we briefly discuss our proposed stepsize rule in this special case of a smooth convex function and provide a geometrical illustration in comparison with the existing rules reviewed above.

Table 1. Summary of the main results of this paper.

Algorithm	Problem	Stepsize rule	Convergence rate
nonaccelerated (Theorem 2.3)	composite	$\phi(2\lambda) \le \phi(\lambda) - \lambda\langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x)\|^2$	$\mathcal{O}(k^{-1})$
nonaccelerated (Corollary 2.6)	smooth	$\phi(2\lambda) \le \phi(\lambda) + \frac{\lambda}{2}\phi'(0)$	$\mathcal{O}(k^{-1})$
accelerated (Theorem 3.1)	composite	$\phi(2\lambda) \le \phi(\lambda) - \lambda\langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x)\|^2$	$\mathcal{O}(k^{-2})$
accelerated (Corollary 3.2)	smooth	$\phi(2\lambda) \le \phi(\lambda) + \frac{\lambda}{2}\phi'(0)$	$\mathcal{O}(k^{-2})$

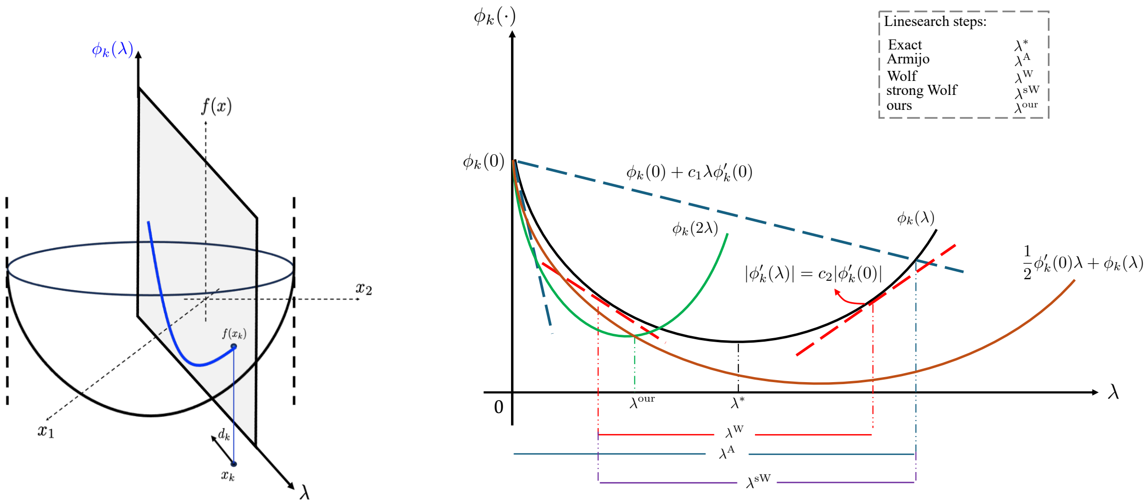
Partial result: If f is a smooth and convex function, our proposed stepsize rule is defined as

$$\lambda_k^{\text{our}} := \arg\max_{\lambda} \left\{ \lambda \mid \phi_k(2\lambda) \le \frac{1}{2}\phi_k'(0)\lambda + \phi_k(\lambda) \right\} \tag{7}$$

where the function ϕ_k is defined in (2) and visualized in Figure 1a. Recall that in the case of non-accelerated methods, the update direction is simply $d_k = -\nabla f(x)$. We emphasize that the proposed linesearch above only requires additional zeroth-order information, namely the objective function evaluation, while the first-order information is the gradient of the function which is already computed in the previous step. Figure 1b illustrates a geometric interpretation of the proposed stepsize rule (7), together with the existing literature reviewed in the preceding section. For further geometrical interpretation, particularly in case of accelerated methods, we refer to [13].

The paper is organized as follows: In Section 2, we present the theoretical results for the non-accelerated adaptive stepsize algorithm. Section 3 provides technical proofs for the accelerated adaptive stepsize case. Some implementation techniques and illustrative examples to show the efficiency of our approaches are presented in Section 4. Finally, the conclusion and further discussion are given in Section 5.

Notation. Let $\mathcal{X}$ be the finite-dimensional real vector space with the standard inner product $\langle\cdot,\cdot\rangle$ and ℓ_p-norm $\|\cdot\|_p$ (by $\|\cdot\|$, we mean the Euclidean standard 2-norm). If f is differentiable, then $\nabla f(x)$ represents the gradient of f at x. The function f is L-smooth, or equivalently the gradient



(a) $\phi_k(\lambda)$. (b) Stepsize rules.

Figure 1. Geometric interpretation of different stepsize rules using $\phi_k(\lambda)$ defined in (2).

of f is L-Lipschitz, if for all $x, y \in \mathbb{R}^d$ one of the following inequalities is satisfied:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \tag{8a}$$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|^2. \tag{8b}$$

Furthermore, f is locally smooth if it is smooth over any compact set of its domain, see [32, Chap. 2] for more details. The prox operator of a convex function $h\colon \mathcal{X} \to \mathbb{R}$ is defined as

$$\operatorname{prox}_h(x) = \arg\min_u h(u) + \frac{1}{2}\|u - x\|^2. \tag{9}$$

A function is "prox-friendly" if the prox operator in (9) is (computationally or explicitly) available. We also denote the gradient mapping of two convex functions f and h by

$$G^f_{\lambda h}(x) := \frac{1}{\lambda}\Big(x - \operatorname{prox}_{\lambda h}(x - \lambda\nabla f(x))\Big), \tag{10}$$

where λ is a positive scalar and has a stepsize interpretation. The gradient mapping is available if f is differentiable and h is prox-friendly.

2. Non-Accelerated Adaptive Stepsize

We present our algorithm and its convergence analysis for the non-accelerated case in this section. From a high-level viewpoint, the proposed rule follows the proximal gradient descent method with the difference in the choice of stepsize λ, which depends on the previous state and its gradient.

2.1. Preliminaries

We first proceed with some assumptions and lemmas that will be used throughout the paper. We note that the classical Cauchy-Schwartz and convexity inequalities are the two useful tools in our analysis and most of the results build on the seminal works by Nesterov [29] and Polyak [37]. The following assumptions holds throughout this study.

Assumption 2.1 (Convex regularity). *The function F in (6) admits bounded level sets, whereby the terms $f(x)$ and $h(x)$ are smooth and prox-friendly, respectively.*

The next lemma indicates several properties of the composite minimization (6) that are central to develop the algorithms in this paper.

Lemma 2.2 (Gradient mapping). *Let $G^f_{\lambda h}(x)$ be the gradient mapping defined in (10) for a smooth convex function f, a possibly nonsmooth function h, and a positive constant λ in $\mathbb{R}_+$.*

(i) **Stationary condition:** *The vector x is a minimizer of (6) if and only if $G^f_{\lambda h}(x) = 0$.*

(ii) **Convexity-like inequality:** *For all x, y in $\mathbb{R}^d$ we have*

$$h\left(x - \lambda G^f_{\lambda h}(x)\right) \leq h(y) - \langle G^f_{\lambda h}(x) - \nabla f(x), y - (x - \lambda G^f_{\lambda h}(x))\rangle.$$

(iii) **Increment bound:** *Considering $x^+ = x - \lambda G^f_{\lambda h}(x)$, for any $z \in \mathbb{R}^d$ we have*

$$F(x^+) - F(z) \leq \langle x^+ - x, \nabla f(x^+) - \nabla f(x) + \frac{1}{2} G^f_{\lambda h}(x)\rangle - \Big(\frac{1}{2\lambda}\|x^+ - x\|^2 + \frac{1}{\lambda}\langle x^+ - x, x - z\rangle\Big).$$

(iv) **Zero-order linsearch:** *Let the update direction $d_k = -G^f_{\lambda h}(x)$. If λ satisfies*

$$\phi_k(2\lambda) \leq \phi_k(\lambda) - \lambda\langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x))\|^2, \tag{11}$$

we then have $\langle x^+ - x, \nabla f(x^+) - \nabla f(x) + \frac{1}{2} G^f_{\lambda h}(x)\rangle \leq 0$.

Before providing the technical proof of the lemma, let us offer some insights into why the above four statements will help us develop algorithms: The convex-like inequality (ii) is a gradient mapping intrinsic characteristic that is particularly useful for controlling the increment of the original function F in (6) along the direction of $d_k = -G^f_{\lambda h}(x)$. In both cases of the non-acceleration and acceleration approaches, the increment inequality (iii) is crucial in proving our convergence analysis. The increment bound (iii) allows for the inclusion of a momentum term that emerges in the acceleration dynamics. The first term on the right-hand side of (iii) is an unfavorable term which is often targeted via the stepsize rule to remain negative. Checking the linesearch in (iv) guarantees the negativity of this unfavorable term, which only requires additional zeroth-order information of function F while the gradient (first-order) information of the gradient mapping $G^f_{\lambda h}(x)$ has already been computed. This is a crucial feature, making a stepsize rule computationally more useful in practice. We conclude this section with the proof of the lemma.

Proof of Lemma 2.2. We provide the proof of each part separately as follows:
Part (i): Assume $\hat{x}$ is the minimizer of the composite minimization (6). Then, by the definition of prox operator (9), we have the following equivalent implications

$$\begin{aligned} G^f_{\lambda h}(\hat{x}) = 0 \iff \hat{x} = \mathrm{prox}_{\lambda h}\left(\hat{x} - \lambda \nabla f(\hat{x})\right) &\iff (\hat{x} - \lambda\nabla f(\hat{x})) - \hat{x} \in \lambda\partial h(\hat{x}) \\ \iff -\lambda\nabla f(\hat{x}) \in \lambda \partial h(\hat{x}) \iff 0 \in \nabla f(\hat{x}) + \partial h(\hat{x}) &\iff \hat{x} \text{ is the minimizer of (6).} \end{aligned}$$

Part (ii): Let $r = \text{prox}_{\lambda h}(t)$. By the definition (9) and the first order optimality condition, we can write

$$r = \arg\min_r h(r) + \frac{1}{2\lambda}||r-t||^2 \iff 0 \in \partial h(r) + \frac{1}{\lambda}(r-t) \iff t - r \in \lambda \partial h(r). \tag{12}$$

By defining $u := x - \lambda G^f_{\lambda h}(x)$ and $w := x - \lambda \nabla f(x)$ we have

$$u = x - \lambda G^f_{\lambda h}(x) = x - \lambda \frac{1}{\lambda}\Big(x - \text{prox}_{\lambda h}(x - \lambda \nabla f(x))\Big) = \text{prox}_{\lambda h}(x - \lambda \nabla f(x)) = \text{prox}(w).$$

Now, using (12), we can conclude: $G^f_{\lambda h}(x) - \nabla f(x) \in \partial h(x - \lambda G^f_{\lambda h}(x))$. Finally, using convexity of h we can write

$$h(x - \lambda G^f_{\lambda h}(x)) \le h(y) - \langle G^f_{\lambda h}(x) - \nabla f(x), y - (x - \lambda G^f_{\lambda h}(x))\rangle.$$

Part (iii): By using the definition of F we have

$$\begin{aligned} F(x^+) - F(z) &= f(x^+) + h(x^+) - f(z) - h(z) \\ &= f(x - \lambda G^f_{\lambda h}(x)) - f(z) + h(x - \lambda G^f_{\lambda h}(x)) - h(z) \end{aligned}$$

Using the result of (ii) on the right-hand side of the above equality and the convexity of F yields

$$\begin{aligned} F(x^+) - F(z) &\le f(x - \lambda G^f_{\lambda h}(x)) - f(x) + \langle \nabla f(x), x - z\rangle - \langle G^f_{\lambda h}(x) - \nabla f(x), z - (x - \lambda G^f_{\lambda h}(x))\rangle \\ &\le \langle \nabla f(x - \lambda G^f_{\lambda h}(x)), x - \lambda G^f_{\lambda h}(x) - x\rangle + \langle G^f_{\lambda h}(x), x^+ - z\rangle + \langle \nabla f(x), \lambda G^f_{\lambda h}(x)\rangle \\ &= \langle x^+ - x, \nabla f(x^+)\rangle - \frac{1}{\lambda}\langle x^+ - x, x^+ - z\rangle + \langle x - x^+, \nabla f(x)\rangle. \end{aligned} \tag{13}$$

By adding and subtracting $\frac{1}{\lambda}\langle x^+ - x, x\rangle$ in (13), we obtain

$$F(x^+) - F(z) \le \langle x^+ - x, \nabla f(x^+) - \nabla f(x) + \frac{1}{2} G^f_{\lambda h}(x)\rangle - \left(\frac{1}{2\lambda}\|x^+ - x\|^2 + \frac{1}{\lambda}\langle x^+ - x, x - z\rangle\right).$$

Part (iv): Replacing the definition $\phi_k(\cdot)$ from (2) in (11) leads to

$$f(x - 2\lambda G^f_{\lambda h}(x)) - f(x - \lambda G^f_{\lambda h}(x)) \le -\lambda \langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x)\|^2.$$

Using the convexity of f on the left-hand side of the above inequality yields

$$\langle \nabla f(x - \lambda G^f_{\lambda h}(x)), x - 2\lambda G^f_{\lambda h}(x) - x + \lambda G^f_{\lambda h}(x)\rangle \le -\lambda \langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x)\|^2.$$

Considering the definition of x^+ in part (iii), we rearrange the above inequality and arrive at

$$\begin{aligned} &\langle \nabla f(x - \lambda G^f_{\lambda h}(x)) - \nabla f(x) + \frac{1}{2} G^f_{\lambda h}(x), G^f_{\lambda h}(x)\rangle \ge 0 \\ \implies &\langle \nabla f(x^+) - \nabla f(x) + \frac{1}{2} G^f_{\lambda h}(x), x^+ - x\rangle \le 0. \end{aligned}$$

□

2.2. Non-accelerated composite minimization

This section focuses on devising an adaptive stepsize rule for non-accelerated algorithms (i.e., $d_k = -G^f_{\lambda h}(x)$) for the general class of composite functions (6). Next, we propose our non-accelerated stepsize rule for the general composite minimization problem (6). In this case, the zero-order linesearch (iv) is the driving force behind the proposed stepsize rule, after which we only need to apply the classic gradient descent update (1) with $d_k = -G^f_{\lambda h}(x)$. This discussion is formalized in the next theorem.

Theorem 2.3 (Non-accelerated adaptive stepsize)**.** *Consider the function F in (6) as a composition of a smooth function $f(x)$ and a prox-friendly function $h(x)$. Assume that the sequence $(x_k)_{k\in\mathbb{N}}$ generated by the update algorithm (1) in which the direction and stepsize rule are*

$$\begin{cases} \lambda_k = \max\left\{\lambda \in \mathbb{R}_+ \mid \phi_k(2\lambda) \le \phi_k(\lambda) - \lambda\langle G^f_{\lambda h}(x_k), \nabla f(x_k)\rangle + \dfrac{\lambda}{2}\|G^f_{\lambda h}(x_k))\|^2\right\}, \\ d_k = -G^f_{\lambda_k h}(x_k). \end{cases} \tag{Alg. 1}$$

Then, we have the uniform stepsize lower bound $\lambda_k \ge 1/2L$ where L is the smoothness constant of f. Moreover, we have the sub-optimality bound $F(x_k) - F(x^*) \le \dfrac{D}{k}$ where D is a constant that only depends on the initial condition x_0, the optimal solution x^*, and the smoothness constant L.

Proof. First, to show that λ_k is bounded from below by $1/2L$, we just need to show that the condition in the stepsize rule always is satisfied by $\lambda_k = 1/2L$. Considering the smoothness of f, we can write the inequality (8b) for the pair (x, x^+) where $x^+ = x - \lambda G^f_{\lambda h}(x)$, which arrives at the two inequalities

$$f(x^+) \le f(x) - \langle \nabla f(x), \lambda G^f_{\lambda h}(x)\rangle + \frac{L\lambda^2}{2}\|G^f_{\lambda h}(x)\|^2,$$
$$f(x) \le f(x^+) + \langle \nabla f(x^+), \lambda G^f_{\lambda h}(x)\rangle + \frac{L\lambda^2}{2}\|G^f_{\lambda h}(x)\|^2.$$

Adding the two sides of the above inequalities leads to

$$\langle G^f_{\lambda h}(x), \nabla f(x^+) - \nabla f(x) + L\lambda G^f_{\lambda h}(x)\rangle \ge 0, \qquad \forall\, \lambda \in \mathbb{R}_+,\, x \in \mathbb{R}^n. \tag{14}$$

which is the same as inequality condition in the stepsize rule when $\lambda = 1/2L$, demonstrating that the stepsize rule condition in Alg. 1 is always satisfied by $\lambda = 1/2L$. Next, we prove the convergence of Alg. 1. By putting $z = x_k$ in the inequality (iii) we have

$$F(x_{k+1}) - F(x_k) \le -\frac{\lambda_k}{2}\|G^f_{\lambda h}(x_k)\|^2 \tag{15}$$

Telescoping the above inequalities from $k = 0$ to $k = \infty$ and using the fact that $\lambda_k \ge 1/2L$ gives us

$$F(x^\infty) - F(x_0) \le -\frac{1}{4L}\sum_{i=0}^{\infty}\|G^f_{\lambda h}(x_i)\|^2$$

According to Assumption 2.1 that $F(x)$ has a finite minimizer, we conclude $\|G^f_{\lambda h}(x^\infty)\|^2 \to 0$ which shows the convergence from (i). To drive the rate of convergence we start from (15) with $z = x^*$.

$$F(x_{k+1}) \le F(x^*) - \frac{1}{2\lambda_k}\|x_{k+1} - x_k\|^2 - \frac{1}{\lambda_k}\langle x_{k+1} - x_k, x_k - x^*\rangle \tag{16}$$

By adding and subtracting $\frac{1}{\lambda_k}\langle x_{k+1}, x_k\rangle$ and $\frac{1}{\lambda_k}\langle x_k, x_k\rangle$ terms and expanding the right-hand side of (16) we have

$$F(x_{k+1}) \leq F(x^*) + \frac{1}{2\lambda_k}\Big(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\Big) \tag{17}$$

Using (17) and the lower bound of λ_k, we can write

$$F(x_{k+1}) \leq F(x^*) + L\Big(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\Big)$$

Summing above inequality from $k = 1$ to $k = T$ gives us

$$\sum_{k=1}^{T}\Big(F(x_k) - F(x^*)\Big) \leq L\Big(\|x_0 - x^*\|^2 - \|x_T - x^*\|^2\Big) \leq L\|x_0 - x^*\|^2$$

Because the sequence of $F(x_i)$ is nondecreasing, we can apply Jensen's inequality to arrive at

$$F(x_k) - F(x^*) \leq \frac{1}{k}\sum_{i=1}^{k}\Big(F(x_i) - F(x^*)\Big) \leq L\Big(\|x_0 - x^*\|^2 - \|x_k - x^*\|^2\Big) \leq \frac{L\|x_0 - x^*\|^2}{k} = \frac{D}{k}.$$

□

Theorem 2 can be extended to a slightly more general setting where f is locally smooth.

Corollary 2.4 (Locally smooth function)**.** *The sequence $(x_k)_{k\in\mathbb{N}}$ generated by (1) for the direction and stepsize rule Alg. 1 ensures the boundedness of sequence and the sub-optimality bound $F(x_k) - F(x^*) \leq \frac{D}{k}$, where f is locally smooth function in (6).*

Proof. This extension essentially indicates that the smoothness condition concerning the term f reflected in the constant L is only required over a compact set. To this end, we recall the proposed algorithm is monotone thanks to the inequality (15). That is, $f(x_{k+1}) \leq f(x_k)$ for all k, implying that the entire sequence of $(x_k)_{k\in\mathbb{N}}$ remains in the sublevel set of $\{x \in \mathbb{R}^n \ : \ f(x) \leq f(x_0)\}$. Note also that the regularity condition in Assumption 2.1 implies that the levelset of the objective function f is compact. This concludes the desired assertion. □

Remark 2.5 (Approximate adaptive stepsize rule)**.** *The stepsize rule Alg. 1 involves a scalar optimization problem. In practice, we use a simple backtracking method to approximate the optimal solution. To this end, we start with a initial λ_0 and scales it with a coefficient $C = (0, 1)$ as*

$$\lambda_k = \max\left\{\lambda_0 C^i \mid \phi_k(2\lambda) \leq \phi_k(\lambda) - \lambda\langle G^f_{\lambda h}(x_k), \nabla f(x_k)\rangle + \frac{\lambda}{2}\|G^f_{\lambda h}(x_k))\|^2,\ i \in \mathbb{N}\right\}.$$

The above approximate solution is guaranteed to satisfy $\lambda_k \geq \frac{C}{2L}$.

2.3. Non-accelerated smooth minimization

In this subsection, we refine the result proposed in the previous subsection for (locally) smooth minimization, i.e., $h(x) = 0$ in (6). Specifically, the analysis in Theorem 2.3 can also be applied to a (locally) smooth minimization. Note that in the smooth case, $G^f_{\lambda h}(x) = \nabla f(x)$ and the linesearch in Alg. 1 is simplified to finding the largest λ satisfying $\phi(2\lambda) \leq \phi(\lambda) + \frac{\lambda}{2}\phi'(0)$. The following corollary explain the results when we invoke Alg. 1 to minimize the (locally) smooth function.

Corollary 2.6 (Convergence of smooth minimization)**.** *Let the function F in (6) be a smooth (i.e., $h(x) = 0$). Then, the stepsize rule in Alg. 1 reduces to (7) while ensuring Theorem 2.3 results of the uniform stepsize lower bound $\lambda_k \geq 1/2L$ and the sub-optimality bound $F(x_k) - F(x^*) \leq \dfrac{D}{k}$.*

Comparison of different stepsizes regime: We close this subsection by highlighting a common feature shared between the Polyak stepsize (4), Armijo stepsize rule (3a), and our proposed stepsize rule (7). These three stepsizes, denoted by λ, fall into the interval

$$\frac{1}{2L} \leq \lambda \leq c\,\frac{f(y_1) - f(y_2)}{\|\nabla f(x_k)\|^2}, \tag{18a}$$

where the points y_1 and y_2 potentially depend on x_k and the stepsize λ. More specifically, we have

$$\begin{cases} \text{Polyak (4)}: & y_1 = x_k, & y_2 & = x^*, & c & = 1, \\ \text{Armijo (3a)}: & y_1 = x_k, & y_2 & = x_k - \lambda\nabla f(x_k), & c & > 1, \\ \text{Our stepsize (7)}: & y_1 = x_k - \lambda\nabla f(x_k), & y_2 & = x_k - 2\lambda\nabla f(x_k), & c & = 2. \end{cases} \tag{18b}$$

The common lower bound $1/2L$ in (18a) is well known for Polyak and Armijo, and it is also shown for our proposed rule in Theorem 2.3. Notice that the optimal choice of the stepsizes is the upper bound of the admissible interval (18a). It is also important to note that in the case of Armijo (3a) and our proposed choice (7), this upper bound also depends on λ (see (18b)), making the stepsize rule implicit. Finally, we wish to draw attention to the choice of the upper bound between Armijo and our proposed approach in (18a): One can inspect that the proposed rule (7) looks one step ahead of Armijo in the direction of ∇f. This may explain why the proposed rule (7) performs numerically better, as it can approximate the curvature of the function f by "looking one step further" than Armijo. Finally, it is also worth noting that the recent work [24] offers an adaptive stepsize rule with the same lower bound $1/2L$ and an explicit upper bound (i.e., independent of λ) that depends on the gradient of two successive iterations.

3. Accelerated Adaptive Stepsize

This section pertains to the analysis of convergence for update direction d_k in accelerated methods. The general form of our accelerated minimization algorithm is provided in Alg. 2, which follows the accelerated proximal gradient descent method with the difference in the choice of stepsize λ [29, 32, 12]. Differently from [29, 8], the stepsize is not fixed, and in our numerical experience, it does not strictly decrease in many iterations, which partially explains the acceleration of the algorithm up to the optimal theoretical bound $\mathcal{O}(k^{-2})$.

Theorem 3.1 (Accelerated adaptive stepsize)**.** *Consider the function F in (6) as a composition of a smooth function $f(x)$ and a prox-friendly function $h(x)$. The sequence $(x_k)_{k\in\mathbb{N}}$ generated by (1) with the direction and stepsize rule*

$$\begin{cases} \beta_k = \dfrac{1 + \sqrt{1 + 4\beta_{k-1}^2}}{2}, \; \gamma_k = \dfrac{1 - \beta_k}{\beta_{k+1}}, \; x_0 = x_1, \; \beta_1 = 0, \\ \lambda_k = \arg\max\limits_{\lambda\in\mathbb{R}} \left\{ \lambda \mid \phi(2\lambda) \leq \phi(\lambda) - \lambda\langle G^f_{\lambda h}(x), \nabla f(x)\rangle + \dfrac{\lambda}{2}\|G^f_{\lambda h}(x))\|^2, \, \lambda \leq \lambda_{k-1} \right\}, \\ d_k = -\gamma_k \left(\dfrac{\lambda_{k-1}}{\lambda_k} d_{k-1} - (1 + \dfrac{1}{\gamma_k}) G^f_{\lambda_k h}(x_k) + \dfrac{\lambda_{k-1}}{\lambda_k} \nabla G^f_{\lambda_{k-1} h}(x_{k-1}) \right). \end{cases} \tag{Alg. 2}$$

ensures the uniform stepsize lower bound $\lambda_k \geq 1/2L$ and the sub-optimality bound $F(x_k) - F(x^*) \leq \frac{D'}{k^2}$ where L is the smoothness constant of f, and D' is a constant that only depends on the initial condition x_0, the optimal solution x^*, and the constant L.

Proof. The proof of a first proposition is the same as in Theorem 2.3. Specifically, according to stepsize rule in Alg. 2, we should first check whether the previous λ_{k-1} satisfies the linesearch condition. Since the inequality (14) holds for $\lambda = 1/2L$ (according to the linesearch condition in Alg. 2), we have the guarantee that $\lambda_k \geq 1/2L$. Regarding the convergence analysis bound, in the first step, we invoke the increment inequality of (iii) in Lemma 2.2 for two cases of $z = y_k$ and $z = x^*$, which arrives at

$$F(y_{k+1}) - F(y_k) \leq -\frac{1}{2\lambda_k}\|y_{k+1} - x_k\|^2 - \frac{1}{\lambda_k}\langle y_{k+1} - x_k, x_k - y_k\rangle, \tag{19a}$$

$$F(y_{k+1}) - F(x^*) \leq -\frac{1}{2\lambda_k}\|y_{k+1} - x_k\|^2 - \frac{1}{\lambda_k}\langle y_{k+1} - x_k, x_k - x^*\rangle. \tag{19b}$$

Let us define $\delta_k := F(y_k) - F(x^*)$. Then, multiplying (19a) by $\beta_k - 1$, and adding the two sides of the inequality to (19b) yields

$$\beta_k\delta_{k+1} - (\beta_k - 1)\delta_k \leq -\frac{\beta_k}{2\lambda_k}\|y_{k+1} - x_k\|^2 - \frac{1}{\lambda_k}\langle y_{k+1} - x_k, \beta_k x_k - (\beta_k - 1)y_k - x^*\rangle.$$

Multiplying the above inequality by $\lambda_k\beta_k$ and considering $\beta_{k-1}^2 := \beta_k^2 - \beta_k$ and using the fact that $\lambda_k \leq \lambda_{k-1}$, we have

$$\lambda_k\beta_k^2\delta_{k+1} - \lambda_{k-1}\beta_{k-1}^2\delta_k \leq -\frac{1}{2}\Big(\|\beta_k(y_{k+1} - x_k)\|^2 + 2\beta_k\langle y_{k+1} - x_k, \beta_k x_k - (\beta_k - 1)y_k - x^*\rangle\Big). \tag{20}$$

The right-hand side of (20) can be equivalently written as

$$\|\beta_k(y_{k+1} - x_k)\|^2 + 2\beta_k\langle y_{k+1} - x_k, \beta_k x_k - (\beta_k - 1)y_k - x^*\rangle = \\ \|\beta_k y_{k+1} - (\beta_k - 1)y_k - x^*\|^2 - \|\beta_k x_k - (\beta_k - 1)y_k - x^*\|^2. \tag{21}$$

Substituting (21) into (20) and rearranging the inequality lead to

$$\lambda_k\beta_k^2\delta_{k+1} - \lambda_{k-1}\beta_{k-1}^2\delta_k \leq -\frac{1}{2}\Big(\|\beta_k y_{k+1} - (\beta_k - 1)y_k - x^*\|^2 - \|\beta_k x_k - (\beta_k - 1)y_k - x^*\|^2\Big). \tag{22}$$

Using the update rule of x_{k+1} on the right-hand side of (22) reduces to

$$\beta_k y_{k+1} - (\beta_k - 1)y_k - x^* = \beta_{k+1}x_{k+1} - (\beta_{k+1} - 1)y_{k+1} - x^*, \tag{23}$$

which is equivalent to

$$x_{k+1} = \frac{(-1 + \beta_k + \beta_{k+1})}{\beta_{k+1}}y_{k+1} + \frac{1 - \beta_k}{\beta_{k+1}}y_k.$$

By combining (22) and (23) and defining $u_k = \beta_k x_k - (\beta_k - 1)y_k - x^*$, we obtain

$$\lambda_k\beta_k^2\delta_{k+1} - \lambda_{k-1}\beta_{k-1}^2\delta_k \leq \frac{1}{2}(\|u_k\|^2 - \|u_{k+1}\|^2). \tag{24}$$

We note that the inequality (24) can be viewed as the so-called Lyapunov function as it indicates that the term $\lambda_k\beta_k^2\delta_{k+1} + \|u_{k+1}\|^2/2$ is monotonically decreasing in each iteration. Summing inequalities (24) from $k = 1$ to $k = T$, one obtains

$$\lambda_T\beta_T^2\delta_{T+1} - \lambda_0\beta_0^2\delta_1 \leq \frac{1}{2}\|u_1\|^2 - \frac{1}{2}\|u_{T+1}\|^2 \leq \frac{1}{2}||u_1||^2.$$

Setting $\beta_0 = 0$ in the above equation yields $\delta_{T+1} \le \frac{\|u_1\|^2}{2\lambda_T \beta_T^2}$. Finally, using Remark 2.5 and the growing rate of $\beta_k \ge k/2$, we deduce the desired result

$$F(y_{T+1}) - F(x^*) = \delta_{T+1} \le \frac{4L\|u_1\|^2}{CT^2} \le \frac{D'}{T^2}.$$

□

We can refine the result proposed in Theorem 3.1 for smooth minimization, where the non-smooth part, h, in (6) is $h(x) = 0$. In this case, $G^f_{\lambda h}(x) = \nabla f(x)$, and the linesearch in the stepsize rule Alg. 2 is simplified to find the largest positive scalar λ that satisfies $\phi(2\lambda) \le \phi(\lambda) + \frac{\lambda}{2}\phi'(0)$. This is formalized in the next Corollary.

Corollary 3.2 (Accelerated smooth minimization). *Let the function F in (6) be a smooth (i.e., $h(x) = 0$). Then, the stepsize rule in Alg. 2 reduces to (7) while ensuring Theorem 3.1 results of the uniform stepsize lower bound $\lambda_k \ge 1/2L$ and the sub-optimality bound $F(x_k) - F(x^*) \le \frac{D'}{k^2}$.*

4. Numerical Results

Let us first start the numerical section with a practical discussion concerning the implementation of the approximate solution in Remark 2.5 for the stepsize rule in Alg. 1. The backtracking in Remark 2.5 uses an initial constant λ_0 to approximate the proposed stepsize rule in Alg. 1. On the one hand, if the initial λ_0 is too large, it takes time to compute the desired stepsize satisfying linesearch condition. On the other hand, a too small λ_0 deteriorates the convergence speed [34]. To address this issue, we approximate the function ϕ in (2) via a quadratic function, denoted by $\widetilde{\phi}$, and suggest the desired λ_0 as its optimizer; see Figure 2 for a visual representation of this approximation (cf. Figure 1b).

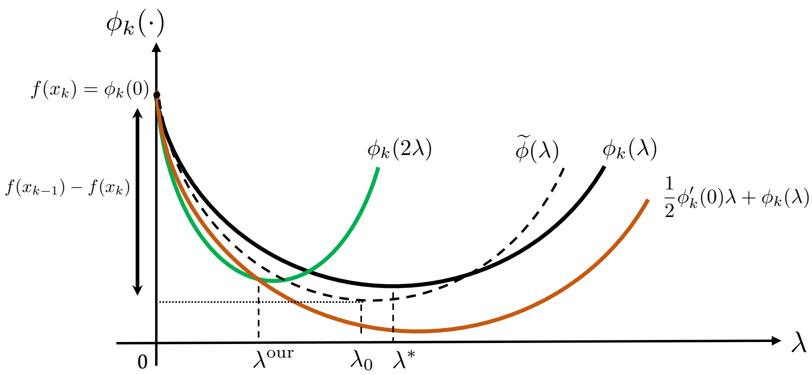


Figure 2. Initial choice of λ_0 at the $(k+1)^{\text{th}}$ iteration.

For the approximate function $\widetilde{\phi}$, we suggest the following criteria:

$$\left.\begin{aligned} &\widetilde{\phi}(0) = \phi(0), \\ &\widetilde{\phi}'(0) = \phi'(0), \\ &\widetilde{\phi}(0) - \min \widetilde{\phi}(\lambda) = f(x_{k-1}) - f(x_k) \end{aligned}\right\} \implies \lambda_0 = \arg\min_{\lambda} \widetilde{\phi}(\lambda) = \frac{2\Big(f(x_{k-1}) - f(x_k)\Big)}{\|\nabla f(x_k)\|^2}.$$

Notice that the function $\widetilde{\phi}$ is constructed in a way whose minimum value is $f(x_{k-1}) - f(x_k)$, i.e., the objective improvement of the last iteration.

We illustrate the performance of our proposed stepsize rules Alg. 1 and Alg. 2 in three cases of (1) smooth minimization including (1-i) logistic regression, (1-ii) quadratic programming, (1-iii) log-sum-exp, and (1-iv) *max-cut* as a representative case of approximate semidefinite programming, (2) composite minimization including (2-i) ℓ_1-Regularized least square, (2-ii) ℓ_1-Constrained least square, (2-iii) ℓ_1-Regularized logistic regression, and (3) non-convex minimization where we consider a cubic objective function. To evaluate our performance, we compare our proposed algorithms with the following methods from the literature: For the class of (1) smooth optimization, we consider the gradient descent (GD) [34], Nesterov's accelerated gradient descent (NAGD) [29], the recent adaptive gradient descent (AdGD) and its heuristic accelerated version (AdGD-accel) [24]. Respectively, for the class of (2) composite minimization, we consider the nonsmooth counterparts including the proximal gradient descent (ISTA) and its accelerated version (FISTA) [8], the adaptive golden ratio algorithm (aGRAAL) [23] and adaptive primal-dual algorithm (APDA) [39]. For (3) non-convex minimization, we use the existing algorithms from the first class of (1) smooth minimization.

4.1. **Smooth minimization**

This section includes four different classes of smooth functions:

(1-i) Logistic regression. Consider the logistic regression problem with quadratic regularization

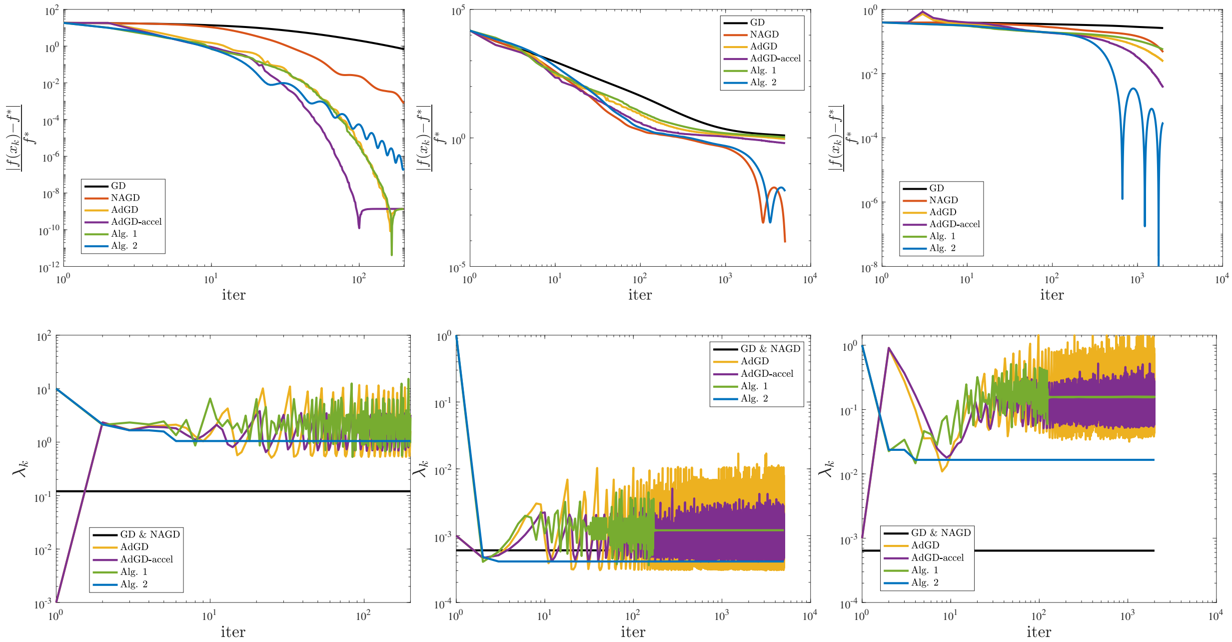
$$\min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} \log\Big(1+\exp(-b_i a_i^\top x)\Big) + \frac{\gamma}{2}\|x\|^2, \tag{25}$$

where N is a number of data, $A = [a_1|a_2|\cdots|a_n] \in \mathbb{R}^{d\times N}$ and $\{b_i\}_{i=1}^N \in \mathbb{R}$ are the features of data, and γ is a regularization parameter proportional to $1/N$. In our experiments, The test data is generated as follows: the matrix $A \in \mathbb{R}^{N\times N}$ is generated randomly using the standard Gaussian distribution $\mathcal{N}(0,1)$, where we generate A with 50% correlated columns as $A(:,j+1) = 0.5A(:,j) + \mathrm{randn}(:)$. The observed measurement vector b is generated as $b := Ax^\natural + \mathcal{N}(0, 0.05)$, where $x^\natural$ is generated randomly using $\mathcal{N}(0,1)$, $d = N = 200$, and we estimate the Lipschitz constant of the gradient by the closed-form expression $L = \|A\|^2/N + \gamma$ used in GD and NAGD as a stepsize. Figure 3a reports the results where all algorithms are initialized at the same point chosen randomly.

(1-ii) Quadratic programming. Consider the quadratic objective function

$$\min_{x\in\mathbb{R}^d} \frac{1}{2}x^\top Bx + b^\top x, \tag{26}$$

where $B \in \mathbb{R}^{d\times d} > 0$ and $b \in \mathbb{R}^d$ are given. This function is smooth with constant $L = \|B\|^2$. The parameters B and b are chosen in the same manner as in the previous part, with the difference that $B = A^\top A$ and b are normalized. Additionally, in our experiments, we set $d = 200$. The results of minimizing this quadratic function are provided in Figure 3b where all the methods are initialized at the same point. It is interesting to note that the proposed stepsize rule for accelerated algorithms exhibits similar behavior to NAGD with a comparative performance while outperforming other adaptive algorithms.



(a) Logistic regression (25) (b) Quadratic programming (26) (c) Log-Sum-Exp (27)

Figure 3. The results for the class (1) smooth minimization. The first row shows the optimality gap, and the second row shows the stepsize behavior.

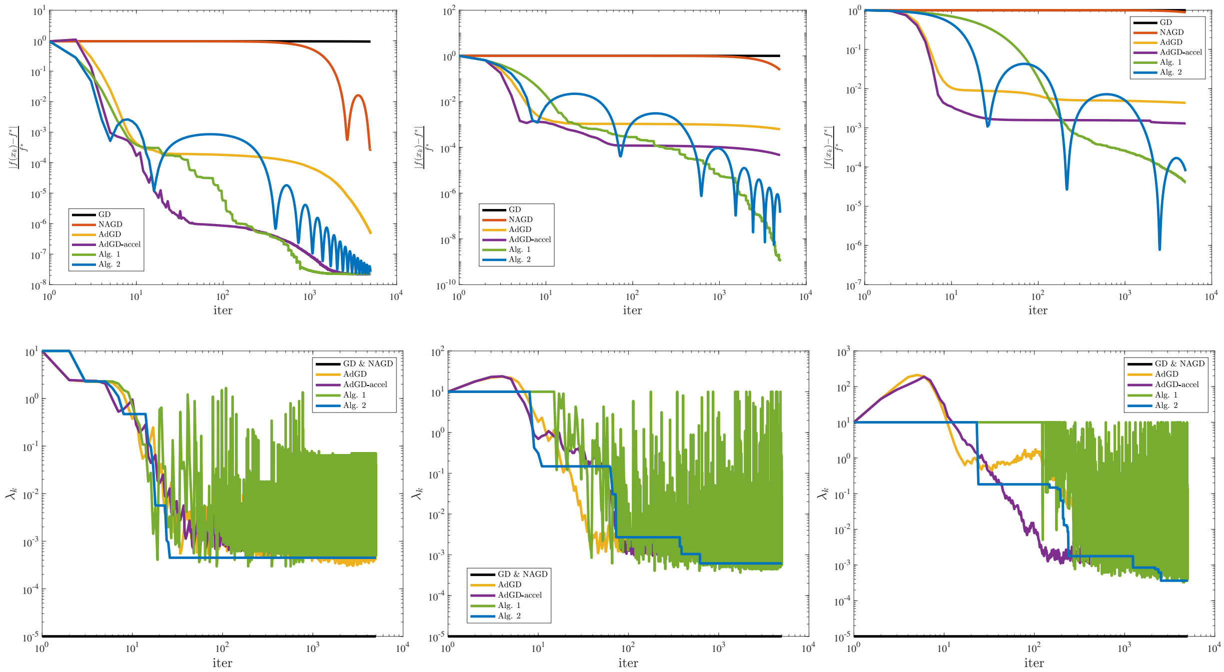
(1-iii) Log-Sum-Exp. Consider the log-sum-exp objective function with regularization

$$\min_{x\in\mathbb{R}^d} \log\left(\sum_{i=1}^{N} \exp(a_i^\top x - b_i)\right) + \frac{\gamma}{2}\|x\|^2, \tag{27}$$

where $A = [a_1|a_2|\cdots|a_n] \in \mathbb{R}^{d\times N}$ and $\{b_i\}_{i=1}^N \in \mathbb{R}$ are the problem data. The log-sum-exp function is viewed as a smooth approximation of $\max_{x\in\mathbb{R}^d}\{a_1^\top x - b_1, a_2^\top x - b_2, \cdots, a_N^\top x - b_N\}$. This function can sometimes be ill-conditioned, making the minimization task particularly difficult for the first-order methods. The function is smooth with the smoothness constant $L = \sigma_{max}(A^\top)$ where $\sigma_{max}(A^\top)$ is a maximum singular value of $A^\top$. In the experiment, we set $N = d = 200$, and A and $\{b_i\}_{i=1}^N$ are generated using a similar approach as in the logistic regression part. The results of minimizing the log-sum-exp function are depicted in Figure 3c where the proposed adaptive accelerated method demonstrates superior efficacy compared to other algorithms.

(1-vi) Approximate semidefinite programming: the case of max-cut. Semidefinite programs are ubiquitous in control problems [11], graph theory [38], network and communication [9]. Most of the problems in this class essentially deal with minimizing the maximum eigenvalue of a matrix, which is a nonsmooth function of its variable. A popular example falling into this category is the *max-cut* problem, whose ε-approximation with the regularizer $\mathcal{R}(x)$ reads as

$$\min_{y\in\mathbb{R}^n} f_\varepsilon(C + \mathrm{diag}(y)) - \langle \mathbf{1}, y\rangle + \eta\mathcal{R}(x), \quad f_\varepsilon(X) = \varepsilon \log\left(\sum_{i=1}^{n} \exp(\lambda_i(X)/\varepsilon)\right), \tag{28}$$



(a) $\eta = 0.1$ (b) $\eta = 0.01$ (c) $\eta = 0.001$

Figure 4. Approximate maximum eigenvalue (28). The first row shows the optimality gap and the second row shows the stepsize behavior.

where C is a constant matrix, η is a regularization coefficient, and $f_\varepsilon(X)$ is the smooth convex approximation of $\lambda_{\mathbf{max}}(X)$. As shown in [30], the gradient of $f_\varepsilon(X)$ can be computed as

$$\nabla f_\varepsilon(X) = \frac{\sum_{i=1}^n \exp(\lambda_i(X)/\varepsilon) q_i q_i^\top}{\sum_{i=1}^n \exp(\lambda_i(X)/\varepsilon)},$$

where q_i is the i^{th} column of the unitary matrix Q in the eigen-decomposition $Q\Sigma Q^\top$ of X with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. In addition, $f_\varepsilon(X)$ is smooth with the smoothness parameter $1/\varepsilon$. Figure 4 shows the simulation results of solving (28) for different regularization coefficient η where the approximation level is $\varepsilon = 10^{-5}$. The matrix C is also generated using the Wishart distribution with $C = G^\top G/\|G\|_2^2$, where G is a standard Gaussian matrix [17], $n = 100$, and $\mathcal{R}(y) = \|y\|^2$.

4.2. Composite minimization

For the case of composite minimization, we consider the following three nonsmooth functions:

(2-i) ℓ_1-Regularized least square.

$$\min_{x\in\mathbb{R}^d} \underbrace{\|Ax - b\|_2^2}_{f(x)} + \underbrace{\gamma\|x\|_1}_{h(x)} \tag{29}$$

(2-ii) ℓ_1-Constrained least square.

$$\min_{x\in\mathbb{R}^d} \{\|Ax - b\|_2^2 : \|x\|_1 \leq 1\} = \min_{x\in\mathbb{R}^d} \underbrace{\|Ax - b\|_2^2}_{f(x)} + \underbrace{\delta_{B_1[0,1]}}_{h(x)} \tag{30}$$

(2-iii) ℓ_1-Regularized logistic regression.

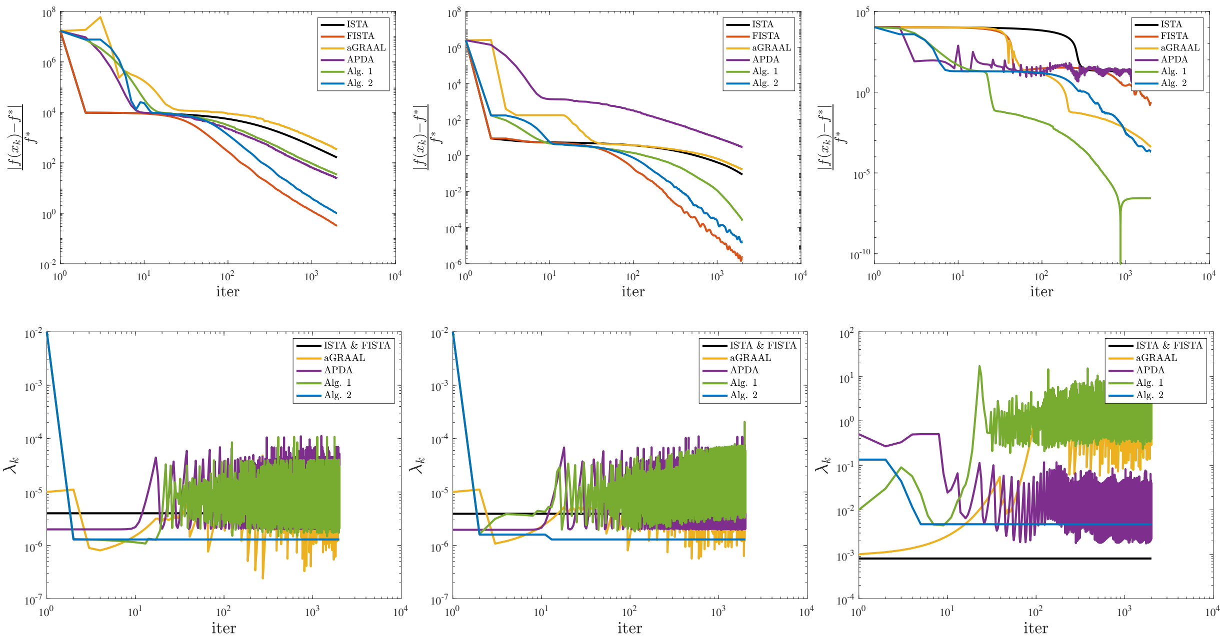
$$\min_{x\in\mathbb{R}^d} \underbrace{\frac{1}{N}\sum_{i=1}^{N}\log\left(1+\exp(-b_i a_i^\top x)\right)}_{f(x)} + \underbrace{\gamma\|x\|_1}_{h(x)} \tag{31}$$

The functions $f(x)$ and $h(x)$ represent the smooth and prox-friendly parts, respectively. In the previous subsection, we explain the smooth parts and their smooth constant. The prox-friendly term $h(x)$ has a closed form solution provided in Table 2 where δ, $B_1[0,1]$, $[\cdot]_+$, and $\odot$ are indicator function, unit ball with 1-norm, positive part selector, and elementwise product, respectively [6].

Table 2. Closed-form prox-operator of $h(x)$.

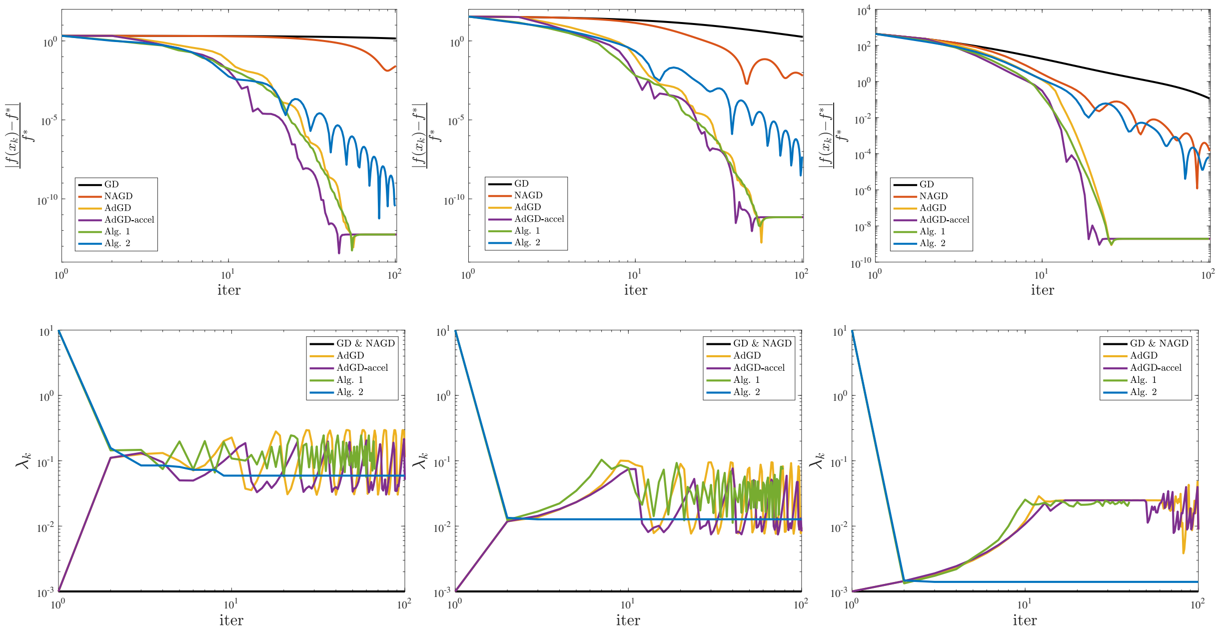
$h(x)$	$\text{prox}_{h(x)}$
$\gamma\|x\|_1$	$\tau_\gamma(x) = [\lvert x\rvert - \gamma\mathbf{e}]_+ \odot \mathbf{sign}(x)$
$\delta_{B_1[0,1]}$	$P_{B_1[0,1]}(x) = \min_{y\in B_1[0,1]} \|y-x\|$

In our simulations, we set $N = d = 200$, γ proportional to $1/N$, $A = 5 \cdot$ `rand(n,n)`, and $b =$ `rand(n,1)`, where `rand(·)` is uniformly distributed random numbers in the interval $(0,1)$. The results are provided in Figure 5. We wish to note that for the objective functions (29) and (30), the proposed accelerated stepsize Alg. 2 has a competitive result and similar behavior with the best performance FISTA [8] while in the case of (31) outperforms all the other algorithms notably.



(a) ℓ_1-Regularized least square (29) (b) ℓ_1-Constrained least square (30) (c) ℓ_1-Regularized logistic reg (31)

Figure 5. The results for the class (2) composite minimization. The first row shows the optimality gap, and the second row shows the stepsize behavior.



(a) $M = 0.5$ (b) $M = 5$ (c) $M = 50$

Figure 6. Cubic regularization problem (32). The first row shows the optimality gap, and the second row shows the stepsize behavior for different values of M.

4.3. Non-convex minimization

We also test our algorithm in non-convex minimization. We consider the case of cubic optimization that is useful in high-order methods [33] and is defined as

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top H x + g^\top x + \frac{M}{6} \|x\|^3. \tag{32}$$

The matrcies $H \in \mathbb{R}^{d \times d}$, $g \in \mathbb{R}^d$, and $M > 0$ are the problem data. Note that in higher-order methods, the constant M has an impact on the convergence rate [31]. Due to the cubic term, the function (32) is neither convex nor smooth. The parameters g and H are the gradient and the Hessian of the logistic loss (25), respectively, and are computed using the same data in part (1-i) logistic regression. We provide the simulations for different M with $d = N = 200$, and the results are reported in Figure 6.

5. Conclusion and Future Directions

This research proposes a new stepsize rule for non-accelerated and accelerated first-order methods for convex composite optimization. The proposed approach provably achieves the convergence optimal rate of $\mathcal{O}(k^{-2})$ for accelerated methods. Compared to other stepsize rules, our proposed scheme is implicit but only requires zeroth-order information to compute a suitable stepsize. Possible extension to this work could be the *min max* problem [26, 41, 10], stochastic gradient descent optimization [21], variational inequalities [27, 19, 25], and non-convex optimization [18, 39, 1]. To mitigate the fact that the stepsize in Alg. 2 is non-increasing, one could re-initialize it finitely many times after a predefined number of iterations. This technique is similar to the restarted accelerated gradient method [35, 20, 40] and it is left as future work.

The authors of a recent work [22] demonstrate that iterations generated by Forward–Backward splitting methods, which include several variants (e.g., ISTA, FISTA), lie on active manifolds within a finite number of iterations and enter a regime of local linear convergence. However, their analysis are based on the standard assumption that the stepsize belongs to $[0, 2/L]$. Departing from this convention is one of the future areas of interest. The authors in [22] also establish and explain why FISTA locally oscillates and can be slower than ISTA under the same standard assumption $\lambda \in [0, 2/L]$. We observe similar behavior in our simulation results (Figures 3a, 5b, 6). Analyzing this behavior for $\lambda > 2/L$, which can also occur in our proposed stepsize rules, could be another future direction.

References

[1] Ahmet Alacaoglu, Yura Malitsky, and Volkan Cevher. Convergence of adaptive algorithms for constrained weakly convex optimization. *Advances in Neural Information Processing Systems*, 34:14214–14225, 2021.

[2] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16, 1966.

[3] Mathieu Barré and Alexandre d'Aspremont. Polyak steps for adaptive fast gradient method. *preprint available at arXiv:1906.03056*, 2019.

[4] Mathieu Barré, Adrien Taylor, and Alexandre d'Aspremont. Complexity guarantees for polyak steps with momentum. In *Conference on learning theory*, pages 452–478, 2020.

[5] Roberto Battiti. First- and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4, 1992.

[6] Amir Beck. *First-order Methods in Optimization*. SIAM, 2017.

[7] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31, 2003.

[8] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2, 2009.

[9] Pratik Biswas, T-C Liang, K-C Toh, Yinyu Ye, and T-C Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE transactions on automation science and engineering*, 3(4):360–371, 2006.

[10] Radu Ioan Boţ, Ernö Robert Csetnek, and Michael Sedlmayer. An accelerated minimax algorithm for convex-concave saddle point problems with nonsmooth coupling function. *Computational Optimization and Applications*, pages 1–42, 2022.

[11] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[12] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[13] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to Nesterov's accelerated gradient descent. *preprint available at arXiv:1506.08187*, 2015.

[14] Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes rendus hebdomadaires des séances de l'Académie des Sciences*, 25, 1847.

[15] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: A novel approach. *Mathematical Programming*, 145, 2014.

[16] Elad Hazan and Sham Kakade. Revisiting the polyak step size. *preprint available at arXiv:1905.00313*, 2019.

[17] Christoph Helmberg and Franz Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[18] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pages 1042–1085, 2018.

[19] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[20] Donghwan Kim and Jeffrey A. Fessler. Adaptive restart of the optimized gradient method for convex optimization. *Journal of Optimization Theory and Applications*, 178, 2018.

[21] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an $mathcalO(1/t)$ convergence rate for the projected stochastic subgradient method. *preprint available at arXiv:1212.2002*, 2012.

[22] Jingwei Liang, Jalal Fadili, and Gabriel Peyré. Activity identification and local linear convergence of forward–backward-type methods. *SIAM Journal on Optimization*, 27(1):408–437, 2017.

[23] Yura Malitsky. Golden ratio algorithms for variational inequalities. *Mathematical Programming*, 184(1-2):383–410, 2020.

[24] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6702–6712, 13–18 Jul 2020.

[25] Yura Malitsky and Matthew K Tam. A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM Journal on Optimization*, 30(2):1451–1472, 2020.

[26] Aryan Mokhtari, Asuman E Ozdaglar, and Sarath Pattathil. Convergence rate of $\mathcal{O}(1/k)$ for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020.

[27] Arkadi Nemirovski. Prox-method with rate of convergence $\mathcal{O}(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[28] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[29] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27, 1983.

[30] Yurii Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2007.

[31] Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[32] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[33] Yurii Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108, 2006.

[34] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[35] Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15, 2015.

[36] B.T. Poliak. *Introduction to Optimization*. Translations series in mathematics and engineering. Optimization Software, Publications Division, 1987.

[37] B. T. Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9, 1969.

[38] Tianyun Tang and Kim-Chuan Toh. Solving graph equipartition sdps on an algebraic variety. *Mathematical Programming*, 204(1):299–347, 2024.

[39] Maria-Luiza Vladarean, Yura Malitsky, and Volkan Cevher. A first-order primal-dual method with adaptivity to local smoothness. *Advances in Neural Information Processing Systems*, 34:6171–6182, 2021.

[40] Tianbao Yang and Qihang Lin. Rsg: Beating subgradient method without smoothness and strong convexity. *The Journal of Machine Learning Research*, 19(1):236–268, 2018.

[41] TaeHo Yoon and Ernest K Ryu. Accelerated algorithms for smooth convex-concave minimax problems with $\mathcal{O}(1/k^2)$ rate on squared gradient norm. In *International Conference on Machine Learning*, pages 12098–12109, 2021.

[42] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International Conference on Machine Learning*, pages 5727–5736, 2018.