OPTIMIZATION ERUDITORUM
Volume 2 (2025), No. 2, 97–115
https://doi.org/10.69829/oper-025-0202-ta02
Tulipa Opera Scholarum

INERTIAL PROXIMAL EXTRAGRADIENT ALGORITHMS FOR NONSMOOTH COMPOSITE OPTIMIZATION

JINTAO YU[1], ZHOU WANG[1], QUN WANG[2], HONGJIN HE[1,*]

[1] *School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China*
[2] *School of Data Sciences, Zhejiang University of Finance and Economics, Hangzhou 310018, China*

Abstract. In this paper, we consider a class of nonsmooth composite optimization problems, where the objective is formed as the sum of a differentiable convex function and a simple nonsmooth convex part. By using the inertial technique, we introduce two improved inertial-type extragradient methods for the problem under consideration. The first one is the doubly-inertial proximal extragradient algorithm (DEGA), which employs two inertial steps to generate intermediate iterates for speeding up the performance of the extragradient method. The second algorithm is called overlapped-inertial proximal extragradient algorithm (OEGA), which utilizes the first inertial step to construct a new inertial step so that more historical information could be used in the final update. With appropriate settings on the inertial parameters, our algorithms can recover the benchmark extragradient method. Theoretically, both DEGA and OEGA are globally convergent under some standard assumptions. Moreover, their effectiveness is verified through some numerical experiments on the Dantzig selector and Lasso problems.

Keywords. Extragradient method, Composite optimization, Proximal gradient method, Inertial, Nonsmooth optimization.

1. Introduction

In this paper, we are interested in a nonsmooth composite optimization problem, which is to minimize the sum of two convex functions, i.e.,

$$\min_{x\in\mathbb{R}^n} \left\{F(x) := f(x) + g(x)\right\}, \tag{1.1}$$

where both $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ and $g(\cdot) : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ are assumed to be closed proper convex functions, and $f(\cdot)$ is further supposed to be differentiable with an L-Lipschitz continuous gradient, while $g(\cdot)$ is allowed to be a nonsmooth function (Noting that $g(\cdot)$ is extended-valued, it is beneficial for encoding constraints on the variable x). Obviously, the model (1.1) is rich enough to recover generic classes of (non)smooth convex optimization problems arising in the areas of compressed sensing, signal and image processing, machine learning, statistical inference, and so on. Here, we refer the reader to monographs (e.g., see [8, 21, 32, 35]) for some specific applications of (1.1).

Considering the split nature of (1.1) and the differentiability of $f(\cdot)$, one of the most popular approaches to finding a solution of (1.1) is the proximal gradient method (PGM, see [4, 5, 33]), which iteratively linearizes the smooth part $f(\cdot)$ at x_k so that the update of x_{k+1} boils down to evaluating a proximal operator, i.e.,

$$x_{k+1} := \mathbf{prox}_{sg}\left(x_k - s\nabla f(x_k)\right), \tag{1.2}$$

[*] Corresponding author.

E-mail address: yujintao0045@163.com (J. T. Yu), wangzhmath@163.com (Z. Wang), wangqun@zufe.edu.cn (Q. Wang), hehongjin@nbu.edu.cn (H. J. He)

2020 Mathematics Subject Classification: 49J53, 49K99.

Accepted: April 5, 2025.

where $s > 0$ is the step size, $\nabla f(x_k)$ is the gradient of $f(\cdot)$ at x_k, and $\mathbf{prox}_{tg}(\cdot)$ is the proximal operator of g defined by

$$\mathbf{prox}_{tg}(\cdot) := \arg\min_{x\in\mathbb{R}^n} \left\{ g(x) + \frac{1}{2t}\|x - \cdot\|^2 \right\}, \quad \forall t > 0. \tag{1.3}$$

In the optimization literature, the method (1.2) is also known as the forward-backward splitting algorithm [12, 13] tailored for maximal monotone inclusions. Moreover, it can also reduce to the classical projected gradient method [6], when $g(\cdot)$ is specified as an indicator function associated with one convex subset of $\mathbb{R}^n$. Due to its simple iterative scheme and the lowest computational cost, the PGM (1.2) received much attention and fruitful applications in machine learning and data science, e.g., see [4, 21, 33, 35] and references therein.

Revisiting the iterative scheme (1.2), we can roughly conclude from earlier studies on this method that there are two bottlenecks motivating researchers to develop more efficient variants. The first one is the comparatively strong requirements on the objective function. As shown in [16, 23], the projected gradient method (a special case of (1.2)) requires $s \in (0, 2\eta/L^2)$ to guarantee its global convergence, where η and L are strongly monotone and Lipschitz continuous moduli of $\nabla f(\cdot)$, respectively. However, the strongly monotone modulus η is difficult, even not impossible to evaluate for many real-world problems. Moreover, such a strong condition does not meet in many cases. As a result, it heavily limits the practicality in applications. To address this issue, Korpelevich [22] judiciously introduced the so-called extragradient method, which takes two sequential gradient steps at each iteration, thereby making it twice as expensive as the standard projected/proximal gradient method. However, the good news is that the extragradient method enjoys promising convergence properties under only the monotonicity and the Lipschitz continuity of $\nabla f(\cdot)$ [26]. Moreover, extragradient methods run much faster than the original projected gradient method in terms of iterations. Along this direction, some improved variants were introduced in the variational inequalities literature, e.g., see [10, 17, 18, 36, 40, 41]. Here, we refer the reader to a most recent survey [37] and references therein.

The second bottleneck of (1.2) is its low convergence behavior in practice. The main reason is related to the aforementioned step size s, which heavily depends on the Lipschitz continuous modulus L of $\nabla f(\cdot)$. Even when L could be easily estimated (e.g., quadratic functions), such a modulus is possibly a large number, thereby resulting a very small step size. Theoretically, it has been documented in [4] that (1.2) has an $O(1/k)$ convergence rate. Therefore, to make an acceleration on (1.2), a simple yet powerful technique is imposing an inertial step to generate an intermediate point for the final update. Such an acceleration technique can be traced back to the seminal work [27] on solving unconstrained minimization with a strongly convex objective. In 2009, Beck and Teboulle [4] further weakened this strong condition and proposed an accelerated PGM enjoying an $O(1/k^2)$ convergence rate for convex composite problem (1.1). More recently, Ochs et al. [29, 30] studied inertial proximal gradient methods for strongly convex and nonconvex (1.1). Their general iterative scheme reads as

$$\begin{cases} y_k := x_k + \alpha_k(x_k - x_{k-1}), & \text{(1.4a)} \\ x_{k+1} := \mathbf{prox}_{sg}\left(y_k - s\nabla f(x_k)\right), & \text{(1.4b)} \end{cases}$$

where $\alpha_k \in [0, 1)$ is an inertial parameter. Clearly, the situation $\alpha_k = 0$ corresponds to the original PGM (1.2). Without the convexity, the $O(1/k^2)$ convergence rate of (1.4) cannot be established for (1.1). However, such an inertial step (i.e., (1.4a)) still greatly speeds up the performance of PGM (1.2). Therefore, in recent years, the inertial technique has been widely used for accelerating many first-order methods, e.g., see [2, 7, 11, 15, 20, 25, 34, 38, 39, 42], to name just a few.

Recently, Nguyen et al. [28] extended the extragradient method [22] to solve (1.1). The iterative scheme takes the form of

$$\begin{cases} y_k := \mathbf{prox}_{sg}\left(x_k - s\nabla f(x_k)\right), & \text{(1.5a)} \\ x_{k+1} := \mathbf{prox}_{\eta g}\left(x_k - \eta\nabla f(y_k)\right), & \text{(1.5b)} \end{cases}$$

where s and η are two positive step sizes. As mentioned above, this method (1.5) possesses nice convergence properties, while it must recall twice evaluations of the proximal operator $\mathbf{prox}_{tg}(\cdot)$ and gradient $\nabla f(\cdot)$, which usually increase the computational burden. Therefore, a natural way of saving the computational cost caused by the extra step (i.e., (1.5a)) is to reduce the number of iterations as many as possible. In this paper, inspired by the efficiency of inertial technique, we introduce two inertial-type proximal extragradient methods to enhance the performance of (1.5). Specifically, we first propose a Doubly-inertial proximal ExtraGradient Algorithm (DEGA), which has two inertial steps generating intermediate iterates. With appropriate settings on the doubly-inertial parameters, our new algorithm recovers the original iterative scheme (1.5). Then, we introduce an Overlapped inertial proximal ExtraGradient Algorithm (OEGA), which also has two inertial steps, but its second one is constructed by the first inertial step. To some extent, the OEGA can absorb more historical information to update the next iterate, which is possibly of benefit for improving the performance of the OEGA. Theoretically, we prove that both proposed algorithms (i.e., DEGA and OEGA) are globally convergent. Finally, some numerical experiments on the Dantzig selector and Lasso problems indicate that our DEGA and OEGA perform well in practice.

The rest of this paper is organized as follows: In Section 2, we recall some basic notations, definitions, and lemmas. In Section 3, we first present the details of the DEGA. Then, we give its global convergence analysis. In Section 4, we introduce the OEGA and analyze its global convergence. In Section 5, some numerical experiments are conducted to verify the reliability of our proposed algorithms. Finally, we complete this paper with drawing some conclusions in Section 6.

2. Preliminaries

In this section, we recall some basic notations, definitions, properties and lemmas that will be used in this paper.

Let $\mathbb{R}^n$ be an n-dimensional Euclidean space endowed with the standard inner product of vectors, i.e., $\langle x, y\rangle = x^\top y$ for any $x, y \in \mathbb{R}^n$, where the superscript $^\top$ represents the transpose of vectors and matrices. Throughout this paper, we let $\|x\| = \sqrt{\langle x, x\rangle}$ be the standard Euclidean norm. Moreover, we denote $\|\cdot\|_1$ and $\|\cdot\|_\infty$ the standard ℓ_1 and ℓ_∞ norms of vectors, respectively. For an extended real-valued function $f(\cdot) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, the domain of f is defined by $\mathbf{dom}\,(f) := \{x \in \mathbb{R}^n \mid f(x) < \infty\}$. The distance from a point $x \in \mathbb{R}^n$ to a nonempty convex set $\Omega \subset \mathbb{R}^n$ is defined by

$$\mathrm{dist}(x, \Omega) := \inf\{\|x - z\| : z \in \Omega\}.$$

Definition 2.1. Let $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a lower semicontinuous convex function. The subdifferential of g is denoted by $\partial g(\cdot) : \mathbb{R}^n \to 2^{\mathbb{R}^n}$, which is given by

$$\partial g(x) := \{\xi \in \mathbb{R}^n \,|\, g(y) \geq g(x) + \langle y - x, \xi\rangle\,,\ \forall y \in \mathbb{R}^n\,\}\,.$$

Then, $\xi \in \partial g(x)$ is called a subgradient of g at point x.

Definition 2.2. Let $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a convex and differentiable function. We say that the gradient $\nabla f(\cdot)$ of f is L-Lipschitz continuous if

$$\|\nabla f(x) - \nabla f(y)\| \leq L\,\|x - y\|\,,\ \forall\, x, y \in \mathbb{R}^n.$$

With the above L-Lipschitz continuity, we recall the well-known decent lemma.

Lemma 2.3 (descent lemma [6]). *Let $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function, and let its gradient ∇f be L-Lipschitz continuous. Then, we have*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2\,,\ \forall\, x, y \in \mathbf{dom}\,(f).$$

Recalling the definition of proximal operator in (1.3), we have the following fundamental inequality.

Lemma 2.4 ([24, Proposition 5.b]). *Let g be a proper closed and convex function. Then, for any $x, y \in \mathbb{R}^n$, the proximal operator $\mathbf{prox}_g(\cdot)$ is nonexpansive, i.e.,*

$$\|\mathbf{prox}_g(x) - \mathbf{prox}_g(y)\| \leq \|x - y\|.$$

Lemma 2.5 ([28]). *Let $u \in \mathbb{R}^n$, $t > 0$, and $v := \mathbf{prox}_{tg}(u)$. Then, for any $w \in \mathbb{R}^n$, we have*

$$g(w) - g(v) \geq \frac{1}{2t}\left(\|u - v\|^2 + \|w - v\|^2 - \|u - w\|^2\right).$$

Below, we recall three useful lemmas that will play important roles in the convergence analysis of this paper.

Lemma 2.6 ([3]). *For any $x, y \in \mathbb{R}^n$, we have the following identities:*

(i). $\|x - y\|^2 = \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle$.
(ii). $\|\gamma x + (1-\gamma)y\|^2 = \gamma\|x\|^2 + (1-\gamma)\|y\|^2 - \gamma(1-\gamma)\|x - y\|^2, \forall\, \gamma \in \mathbb{R}$.

Lemma 2.7 ([1]). *Let $\{\phi_k\}$, $\{\delta_k\}$ and $\{\alpha_k\}$ be the sequences in $[0, +\infty)$ such that $\phi_{k+1} \leq \phi_k + \alpha_k(\phi_k - \phi_{k-1}) + \delta_k$, $\forall k \in \mathbb{N}$, $\sum_{k=1}^{\infty}\delta_k < +\infty$, and there exists a real number $\widehat{\alpha}$ satisfying $0 \leq \alpha_k \leq \widehat{\alpha} < 1$, $\forall k \in \mathbb{N}$. Then, the following properties hold*

(i). $\sum_{k=1}^{\infty}[\phi_{k+1} - \phi_k]_+ < +\infty$, *where* $[a]_+ = \max\{a, 0\}$ *for any* $a \in \mathbb{R}$.
(ii). *there exists* $\phi^\star \in [0, +\infty)$ *such that* $\lim_{k\to\infty}\phi_k = \phi^\star$.

Lemma 2.8 ([31]). *Let $\{x_k\}$ be a sequence such that there exists a nonempty set $\Omega \subset \mathbb{R}^n$ verifying:*

(i). *For every* $x^* \in \Omega$, $\lim_{k\to\infty}\|x_k - x^*\|$ *exists.*
(ii). *If* $\{x_{k_j}\}$ *converges to* $x^\star \in \mathbb{R}^n$ *for a subsequence* $k_j \to \infty$, *then* $x^\star \in \Omega$.

Then, there exists $x^\star \in \Omega$ such that the sequence $\{x_k\}$ converges to $x^\star$ in $\mathbb{R}^n$.

3. Doubly-Inertial Proximal Extragradient Algorithm

In the section, we first present the new inertial-type extragradient algorithm for (1.1), which is named as doubly-inertial proximal extragradient algorithm (DEGA). Then, we establish its global convergence.

3.1. **Algorithmic framework.** As mentioned in Section 1, the ideas of extragradient method (i.e., one more proximal step (1.5a)) and inertial step (i.e., (1.4a)) are able to improve the performance of PGM (1.2). Taking a look at (1.5), we can see that the evaluation on y_k comes from the information generated by x_k, while the update of x_{k+1} is due to the information x_k and $\nabla f(y_k)$. Therefore, we are motivated to employ different iterates instead of the iterate x_k in (1.5). Specifically, we construct two different inertial steps to generate intermediate iterates for updating y_k and x_{k+1} in (1.5a) and (1.5b), respectively. So, we call our algorithm Doubly-inertial ExtraGradient Algorithm (DEGA), which is shown in Algorithm 1.

Algorithm 1 Doubly-inertial proximal extragradient algorithm (DEGA).

```
1: Choose initial points $x_0$ and $x_1 \in \mathbb{R}^n$. Select $\alpha_0, \beta_0, s, \eta$ satisfying Assumptions 1-2.
2: for $k = 1, 2, \cdots$ do
3:
       $w_k = x_k + \alpha_k(x_k - x_{k-1})$,            (3.1a)
       $v_k = x_k + \beta_k(x_k - x_{k-1})$,             (3.1b)
       $y_k = \mathbf{prox}_{sg}(w_k - s\nabla f(v_k))$,       (3.1c)
       $x_{k+1} = \mathbf{prox}_{\eta g}(w_k - \eta\nabla f(y_k))$.   (3.1d)
4: end for
```

Remark 3.1. Obviously, when $\alpha_k = \beta_k = 0$, $s = \eta$, and $g(x)$ is specifying as an indicator function, our algorithm immediately reduces to the original extragradient method [22]. When $g(x)$ is a general nonsmooth convex function, Algorithm 1 corresponds to the extended extragradient method in [28]. When $\alpha_k = \beta_k \neq 0$, we can obtain one-step inertial extragradient method. To a certain extend, Algorithm 1 is more flexible due to the four parameters α_k, β_k, s and η. We shall mention that both s and η could be selected via some dynamical (self-adaptive or line-search) strategies (e.g., see [4, 17, 40, 41]). Here, we assume s and η are constants for simplicity. Indeed, we will show in Section 5 that Algorithm 1 without those dynamical strategies still works better than the original extragradient method and PGM.

3.2. **Convergence analysis.** In this subsection, we show that Algorithm 1 for (1.1) is globally convergent. To begin our analysis, we first recall a pivot lemma that has been established in [28], and we skip its detailed proof here for the conciseness of this paper.

Lemma 3.2. *Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, $t > 0$, and $p = \mathbf{prox}_{tg}(x - t\nabla f(y))$, where f and g are given in* (1.1). *Then, for any $z \in \mathbb{R}^n$, we have*

$$F(z) - F(p) \geq \frac{1}{2t}\left(\|x-p\|^2 + \|z-p\|^2 - \|x-z\|^2\right) - \frac{L}{2}\|p-y\|^2,$$

where L is the Lipschitz continuous constant of $\nabla f(\cdot)$.

Below, we first make some assumptions for the coming global convergence of Algorithm 1.

Assumption 1. *The step sizes s and η satisfy $0 < s \leq \min(\eta, 1/L)$ and $2L\eta \leq 1 - L^2 s^2$.*

Assumption 2. *The inertial sequence $\{\alpha_k\}$ is non-decreasing and $\{\beta_k\}$ satisfy $0 < \alpha_k \leq \widehat{\alpha} < 1$ and $0 < \widetilde{\beta} \leq \beta_k < 1$. Moreover, for some $\bar{\delta} > 0$ and $\bar{\sigma} > 0$, we assume these parameters satisfying*

$$(\widehat{\alpha} + \bar{\delta})\left(\widehat{\alpha}(1+\widehat{\alpha}) + 2L\eta\left(\frac{Ls(\widehat{\alpha} - \widetilde{\beta})}{1 - Ls}\right)^2 + \upsilon\widehat{\alpha}\bar{\delta} + \bar{\sigma}\right) \leq \bar{\delta}\upsilon,$$

where

$$\upsilon := 1 - 2L\eta\left(\frac{1}{1-Ls} - \frac{s}{\eta}\right)^2. \tag{3.2}$$

For the inequality in Assumption 2, we present a lemma to show the positiveness of υ for the rationality of this assumption.

Lemma 3.3. *Suppose that Assumption 1 holds. We then have υ defined in* (3.2) *is positive.*

Proof. Firstly, we rewrite (3.2) that

$$\begin{aligned}\upsilon :&= 1 - 2L\eta\left(\frac{1}{1-Ls} - \frac{s}{\eta}\right)^2 \\ &= \frac{\eta(1-Ls)^2 - 2L\eta^2 - 2Ls^2(1-Ls)^2 + 4Ls\eta(1-Ls)}{\eta(1-Ls)^2}.\end{aligned} \tag{3.3}$$

Since $0 < s \leq \min\left(\eta, \frac{1}{L}\right)$, it is obvious that verifying the positiveness of υ amounts to checking that the numerator of (3.3) is positive, i.e.,

$$\eta(1-Ls)^2 - 2L\eta^2 - 2Ls^2(1-Ls)^2 + 4Ls\eta(1-Ls) > 0,$$

where the left-hand term is a quadratic function with respect to η. Therefore, we need to prove that

$$\begin{cases} \eta < (1-Ls)\dfrac{1 + 3Ls + \sqrt{(1+3Ls)^2 - 16L^2s^2}}{4L}, \\ \eta > (1-Ls)\dfrac{1 + 3Ls - \sqrt{(1+3Ls)^2 - 16L^2s^2}}{4L}. \end{cases}$$

Apparently, if $0 \le b \le a$ then $a - b \le \sqrt{a^2 - b^2}$. Thus, we derive

$$1 - Ls = (1 + 3Ls) - 4Ls \le \sqrt{(1 + 3Ls)^2 - 16L^2s^2}. \tag{3.4}$$

It follows from (3.4) that

$$(1 - Ls)\frac{1 + 3Ls - \sqrt{(1 + 3Ls)^2 - 16L^2s^2}}{4L} \le (1 - Ls)s < \eta$$

and

$$(1 - Ls)\frac{1 + 3Ls + \sqrt{(1 + 3Ls)^2 - 16L^2s^2}}{4L} \ge \frac{(1 - Ls)(1 + Ls)}{2L} > \eta.$$

We complete the proof. □

Now, we show the difference between y_k and x_{k+1} generated by the two proximal steps (i.e., (3.1c) and (3.1d)) satisfying an inequality.

Lemma 3.4. *Let $\{w_k\}, \{v_k\}, \{x_k\}, \{y_k\}$ be sequences generated by Algorithm 1. Suppose that Assumption 1 holds. Then, for any $k \in \mathbb{N}$, it holds that*

$$\|x_{k+1} - y_k\| \le \left(1 - \frac{s}{\eta} + \frac{Ls}{1 - Ls}\right)\|w_k - x_{k+1}\| + \frac{Ls}{1 - Ls}\|w_k - v_k\|. \tag{3.5}$$

Proof. First, we denote

$$z_{k+1} = \mathbf{prox}_{sg}(w_k - s\nabla f(y_k)).$$

Since ∇f is L-Lipschitz continuous, it follows from the nonexpansiveness of $\mathbf{prox}_{tg}(\cdot)$ (see Lemma 2.4) that

$$\begin{aligned}\|y_k - z_{k+1}\| &\le \|w_k - s\nabla f(v_k) - (w_k - s\nabla f(y_k))\| \\ &\le Ls\|y_k - v_k\|.\end{aligned}$$

Consequently, we have

$$\begin{aligned}\|v_k - w_k\| + \|w_k - z_{k+1}\| &\ge \|v_k - z_{k+1}\| \\ &\ge \|v_k - y_k\| - \|y_k - z_{k+1}\| \\ &\ge (1 - Ls)\|y_k - v_k\|. \end{aligned}\tag{3.6}$$

It follows from the optimality condition of (3.1d) that

$$\frac{w_k - x_{k+1}}{\eta} - \nabla f(y_k) \in \partial g(x_{k+1}),$$

which, together with the convexity of g, implies

$$\left\langle \frac{w_k - x_{k+1}}{\eta} - \nabla f(y_k), z_{k+1} - x_{k+1} \right\rangle \le g(z_{k+1}) - g(x_{k+1}). \tag{3.7}$$

Similarly, by invoking the definition of z_{k+1} at the beginning of this proof, we have

$$\left\langle \frac{w_k - z_{k+1}}{s} - \nabla f(y_k), x_{k+1} - z_{k+1} \right\rangle \le g(x_{k+1}) - g(z_{k+1}). \tag{3.8}$$

Adding (3.7) and (3.8) immediately leads to

$$\left\langle \frac{w_k - z_{k+1}}{s} - \frac{w_k - x_{k+1}}{\eta}, w_k - z_{k+1} - (w_k - x_{k+1}) \right\rangle \le 0. \tag{3.9}$$

Reorganizing (3.9), it follows from the Cauchy-Schwarz inequality that

$$\begin{aligned}\frac{\|w_k - z_{k+1}\|^2}{s} + \frac{\|w_k - x_{k+1}\|^2}{\eta} &\le \left(\frac{1}{s} + \frac{1}{\eta}\right) \langle w_k - z_{k+1}, w_k - x_{k+1}\rangle \\ &\le \left(\frac{1}{s} + \frac{1}{\eta}\right) \|w_k - z_{k+1}\|\|w_k - x_{k+1}\|,\end{aligned}$$

which can be rewritten as

$$(\|w_k - z_{k+1}\| - \|w_k - x_{k+1}\|)\left(\|w_k - z_{k+1}\| - \frac{s}{\eta}\|w_k - x_{k+1}\|\right) \le 0, \tag{3.10}$$

which means

$$\|w_k - z_{k+1}\| \le \|w_k - x_{k+1}\| \quad \text{and} \quad \|w_k - z_{k+1}\| \ge \frac{s}{\eta}\|w_k - x_{k+1}\|$$

or

$$\|w_k - z_{k+1}\| \ge \|w_k - x_{k+1}\| \quad \text{and} \quad \|w_k - z_{k+1}\| \le \frac{s}{\eta}\|w_k - x_{k+1}\|.$$

Invoking the requirement $\frac{s}{\eta} \le 1$ on the above two cases, inequality (3.10) implies that

$$\frac{s}{\eta}\|w_k - x_{k+1}\| \le \|w_k - z_{k+1}\| \le \|w_k - x_{k+1}\|. \tag{3.11}$$

Recalling the definitions of y_k and x_{k+1} in (3.1c) and (3.1d), respectively, we can similarly obtain

$$\left\langle \frac{w_k - y_k}{s} - \nabla f(v_k), x_{k+1} - y_k \right\rangle \le g(x_{k+1}) - g(y_k) \tag{3.12}$$

and

$$\left\langle \frac{w_k - x_{k+1}}{\eta} - \nabla f(y_k), y_k - x_{k+1} \right\rangle \le g(y_k) - g(x_{k+1}). \tag{3.13}$$

As a consequence, summing (3.12) and (3.13) yields

$$\frac{1}{s}\|x_{k+1} - y_k\|^2 + \left(\frac{1}{s} - \frac{1}{\eta}\right)\langle x_{k+1} - y_k, w_k - x_{k+1}\rangle \le \langle x_{k+1} - y_k, \nabla f(v_k) - \nabla f(y_k)\rangle,$$

which, together with the Cauchy-Schwarz inequality, implies that

$$\frac{1}{s}\|x_{k+1} - y_k\|^2 \le \left(\frac{1}{s} - \frac{1}{\eta}\right)\|x_{k+1} - y_k\|\|w_k - x_{k+1}\| + \|x_{k+1} - y_k\|\|\nabla f(v_k) - \nabla f(y_k)\|.$$

Using the Lipschitz continuity of ∇f in the above inequality, we immediately have

$$\|x_{k+1} - y_k\| \le \left(1 - \frac{s}{\eta}\right)\|w_k - x_{k+1}\| + Ls\|v_k - y_k\|. \tag{3.14}$$

Combining (3.6), (3.11) and (3.14), we conclude

$$\|x_{k+1} - y_k\| \le \left(1 - \frac{s}{\eta} + \frac{Ls}{1 - Ls}\right)\|w_k - x_{k+1}\| + \frac{Ls}{1 - Ls}\|w_k - v_k\|.$$

The proof is complete. □

Hereafter, we show that the distance between x_{k+1} and x^* satisfies an inequality, where x_{k+1} is given by (3.1d) and x^* is a solution of (1.1).

Lemma 3.5. *Let $\{w_k\}, \{v_k\}, \{x_k\}, \{y_k\}$ be sequences generated by Algorithm 1 and x^* be a solution of (1.1). Suppose that Assumption 1 holds. Then, we have*

$$\begin{aligned}\|x_{k+1}-x^*\|^2 \le&\|w_k-x^*\|^2-\left(1-2L\eta\left(1-\frac{s}{\eta}+\frac{Ls}{1-Ls}\right)^2\right)\|w_k-x_{k+1}\|^2\\&+2L\eta\left(\frac{Ls}{1-Ls}\right)^2\|w_k-v_k\|^2.\end{aligned}\tag{3.15}$$

Proof. Let x^* be a solution of (1.1). First, an application of Lemma 3.2 by setting $x=w_k$, $y=y_k$, $p=x_{k+1}$, $z=x^*$, and $t=\eta$ immediately yields

$$\frac{1}{2\eta}\left(\|w_k-x_{k+1}\|^2+\|x_{k+1}-x^*\|^2-\|w_k-x^*\|^2\right)-\frac{L}{2}\|x_{k+1}-y_k\|^2\le 0,$$

which is due to the fact $F(x^*)-F(x_{k+1})\le 0$, and can be further rewritten as

$$\|x_{k+1}-x^*\|^2\le\|w_k-x^*\|^2-\|w_k-x_{k+1}\|^2+L\eta\|x_{k+1}-y_k\|^2.\tag{3.16}$$

By (3.5), an application of the fact that $(a+b)^2\le 2a^2+2b^2$ for all $a,b\in\mathbb{R}$ to (3.16) yields

$$\begin{aligned}\|x_{k+1}-x^*\|^2 \le&\|w_k-x^*\|^2-\left(1-2L\eta\left(1-\frac{s}{\eta}+\frac{Ls}{1-Ls}\right)^2\right)\|w_k-x_{k+1}\|^2\\&+2L\eta\left(\frac{Ls}{1-Ls}\right)^2\|w_k-v_k\|^2.\end{aligned}$$

The proof is complete. □

With the above preparations, we are now at the stage of stating the global convergence of Algorithm 1.

Theorem 3.6. *Suppose that Assumptions 1 and 2 hold. Then, the sequence $\{x_k\}$ generated by Algorithm 1 converges to a solution of (1.1).*

Proof. It follows from Lemma 2.6 and (3.1a) that

$$\begin{aligned}\|w_k-x^*\|^2=&(1+\alpha_k)\|x_k-x^*\|^2-\alpha_k\|x_{k-1}-x^*\|^2\\&+\alpha_k(1+\alpha_k)\|x_k-x_{k-1}\|^2.\end{aligned}\tag{3.17}$$

Plugging (3.17) into (3.15), we obtain

$$\begin{aligned}&\|x_{k+1}-x^*\|^2-(1+\alpha_k)\|x_k-x^*\|^2+\alpha_k\|x_{k-1}-x^*\|^2\\&\le\left(\alpha_k(1+\alpha_k)+2L\eta\left(\frac{Ls(\alpha_k-\beta_k)}{1-Ls}\right)^2\right)\|x_k-x_{k-1}\|^2-\upsilon\|w_k-x_{k+1}\|^2,\end{aligned}\tag{3.18}$$

where υ is given in (3.2). Similarly, by taking some $\bar{\delta}>0$ such that Assumption 2 holds, it follows from the definition of w_k that

$$\begin{aligned}\|x_{k+1}-w_k\|^2=&\|x_{k+1}-(x_k+\alpha_k(x_k-x_{k-1}))\|^2\\=&\|x_k-x_{k+1}\|^2+\alpha_k^2\|x_k-x_{k-1}\|^2+2\alpha_k\langle x_k-x_{k+1},x_k-x_{k-1}\rangle\\\ge&\|x_k-x_{k+1}\|^2+\alpha_k^2\|x_k-x_{k-1}\|^2\\&+\alpha_k\left(-\frac{1}{\alpha_k+\bar{\delta}}\|x_k-x_{k+1}\|^2-(\alpha_k+\bar{\delta})\|x_k-x_{k-1}\|^2\right)\\=&\frac{\bar{\delta}}{\alpha_k+\bar{\delta}}\|x_k-x_{k+1}\|^2-\bar{\delta}\alpha_k\|x_k-x_{k-1}\|^2.\end{aligned}\tag{3.19}$$

Then, substituting (3.19) into (3.18) yields

$$\begin{aligned}&\|x_{k+1}-x^*\|^2-(1+\alpha_k)\|x_k-x^*\|^2+\alpha_k\|x_{k-1}-x^*\|^2\\&\leq -\frac{\bar{\delta}\upsilon}{\alpha_k+\bar{\delta}}\|x_{k+1}-x_k\|^2+\gamma_k\|x_k-x_{k-1}\|^2,\end{aligned}\tag{3.20}$$

where

$$\gamma_k=\alpha_k(1+\alpha_k)+2L\eta\left(\frac{Ls(\alpha_k-\beta_k)}{1-Ls}\right)^2+\upsilon\alpha_k\bar{\delta}>0.\tag{3.21}$$

By the upper boundedness of $\{\alpha_k\}$ and the lower boundedness of $\{\beta_k\}$, it is not difficult to verify that the γ_k defined by (3.21) has an upper bound $\widehat{\gamma}$, i.e.,

$$\gamma_k\leq\widehat{\gamma}:=\widehat{\alpha}(1+\widehat{\alpha})+2L\eta\left(\frac{Ls(\widehat{\alpha}-\widetilde{\beta})}{1-Ls}\right)^2+\upsilon\widehat{\alpha}\bar{\delta}.\tag{3.22}$$

Below, we first define:

$$\mu_k:=\|x_k-x^*\|^2-\alpha_k\|x_{k-1}-x^*\|^2+\gamma_k\|x_k-x_{k-1}\|^2.$$

Since $\{\alpha_k\}$ is non-decreasing, we have

$$\begin{aligned}\mu_{k+1}-\mu_k=&\|x_{k+1}-x^*\|^2-(1+\alpha_{k+1})\|x_k-x^*\|^2+\alpha_k\|x_{k-1}-x^*\|^2\\&+\gamma_{k+1}\|x_{k+1}-x_k\|^2-\gamma_k\|x_k-x_{k-1}\|^2\\\leq&\|x_{k+1}-x^*\|^2-(1+\alpha_k)\|x_k-x^*\|^2+\alpha_k\|x_{k-1}-x^*\|^2\\&+\gamma_{k+1}\|x_{k+1}-x_k\|^2-\gamma_k\|x_k-x_{k-1}\|^2\\=&\left(\gamma_{k+1}-\frac{\bar{\delta}\upsilon}{\alpha_k+\bar{\delta}}\right)\|x_{k+1}-x_k\|^2.\end{aligned}\tag{3.23}$$

By Assumption 2 and the boundedness of γ_k and α_k, we claim that there exists some $\bar{\sigma}>0$ such that

$$\gamma_{k+1}-\frac{\bar{\delta}\upsilon}{\alpha_k+\bar{\delta}}\leq-\bar{\sigma}<0,\quad\forall k\geq 1,\tag{3.24}$$

which implies

$$(\alpha_k+\bar{\delta})(\gamma_{k+1}+\bar{\sigma})\leq(\widehat{\alpha}+\bar{\delta})(\widehat{\gamma}+\bar{\sigma})\leq\bar{\delta}\upsilon.\tag{3.25}$$

Therefore, we obtain from (3.23) and (3.24) that

$$\mu_{k+1}-\mu_k\leq-\bar{\sigma}\|x_{k+1}-x_k\|^2,\tag{3.26}$$

which implies that the sequence $\{\mu_k\}$ is non-increasing. Hence, it follows from (3.26)and the boundedness of $\{\alpha_k\}\subset(0,\widehat{\alpha}]$ that

$$\|x_k-x^*\|^2-\alpha_k\|x_{k-1}-x^*\|^2\leq\mu_k\leq\mu_1.\tag{3.27}$$

Consequently, it comes from (3.27) that

$$\|x_k-x^*\|^2\leq\widehat{\alpha}^k\|x_0-x^*\|^2+\mu_1\sum_{n=0}^{k-1}\widehat{\alpha}^n\leq\widehat{\alpha}^k\mu_0+\frac{\mu_1}{1-\widehat{\alpha}}\tag{3.28}$$

with $\mu_0=\|x_0-x^*\|^2$. Combining (3.26)-(3.28) immediately leads to

$$\bar{\sigma}\sum_{n=1}^{k}\|x_{n+1}-x_n\|^2\leq\mu_1-\mu_{k+1}\leq\mu_1+\widehat{\alpha}\|x_k-x^*\|^2\leq\widehat{\alpha}^{k+1}\mu_0+\frac{\mu_1}{1-\widehat{\alpha}},$$

which shows that

$$\sum_{k=1}^{\infty}\|x_{k+1}-x_k\|^2<+\infty\tag{3.29}$$

and $\|x_{k+1} - x_k\| \to 0$. It then follows from (3.19) and (3.29) that

$$\|w_k - x_{k+1}\|^2 = \|x_k - x_{k+1}\|^2 + \alpha_k^2 \|x_k - x_{k-1}\|^2 + 2\alpha_k \langle x_k - x_{k+1}, x_k - x_{k-1}\rangle \to 0. \tag{3.30}$$

Moreover, by the definitions of w_k and v_k in (3.1a) and (3.1b), respectively, we have

$$\|w_k - v_k\| \equiv |\alpha_k - \beta_k| \|x_k - x_{k-1}\| \to 0, \tag{3.31}$$

and

$$\|x_k - v_k\| \equiv \beta_k \|x_k - x_{k-1}\| \to 0. \tag{3.32}$$

Hence, from (3.5), (3.30)-(3.31), we get

$$\|x_{k+1} - y_k\| \to 0,$$

which further implies that

$$\|w_k - y_k\| \leq \|w_k - x_{k+1}\| + \|x_{k+1} - y_k\| \to 0 \tag{3.33}$$

and

$$\|x_k - y_k\| \leq \|x_k - x_{k+1}\| + \|x_{k+1} - y_k\| \to 0. \tag{3.34}$$

Finally, we use Lemma 2.8 to show that the sequence $\{x_k\}$ converges to a solution of (1.1). We have proven that inequality (3.20) holds for an arbitrary optimal solution x^*. By (3.20), (3.29) and Lemma 2.7, we derive that $\lim_{k\to\infty} \|x_k - x^*\|$ exists.

On the other hand, let $x^\star$ be a cluster point of a subsequence $\{x_{k_j}\}$ of the sequence $\{x_k\}$, that is, $x_{k_j} \to x^\star$ as $j \to \infty$. From (3.32) and (3.34), it follows that $\{w_k\}$, $\{v_k\}$ and $\{y_k\}$ converges to $x^\star$. We will show that $x^\star$ is a solution of (1.1). First, for every optimal solution x^* of (1.1), it follows from the optimality condition that

$$0 \in \nabla f(x^*) + \partial g(x^*) \iff -\nabla f(x^*) \in \partial g(x^*),$$

which, together with the convexity of g, equals to

$$g(y) \geq g(x^*) - \langle \nabla f(x^*), y - x^*\rangle, \quad \forall y \in \mathbb{R}^n. \tag{3.35}$$

Recalling the definition of y_k and taking the corresponding subsequence $\{y_{k_j}\}$, it follows from the convexity of g that

$$\left\langle \frac{w_{k_j} - y_{k_j}}{s} - \nabla f(v_{k_j}), y - y_{k_j} \right\rangle \leq g(y) - g(y_{k_j}), \quad \forall y \in \mathbb{R}^n. \tag{3.36}$$

Taking the limit as $j \to \infty$ in (3.36), it follows from (3.33) that

$$\left\langle \frac{x^\star - x^\star}{s} - \nabla f(x^\star), y - x^\star \right\rangle \leq g(y) - g(x^\star), \quad \forall y \in \mathbb{R}^n,$$

which, together with (3.35), immediately concludes that the cluster point of $\{x_k\}$ is a solution of (1.1). □

4. Overlapped Inertial Proximal Extragradient Algorithm

In this section, we further propose a variant of Algorithm 1 for (1.1). Also, we will prove its global convergence.

4.1. **Algorithmic framework.**

Revisiting the iterative schemes of Algorithm 1, we observe that both inertial steps use the information of iterates x_k and x_{k-1}, and the only distinction is their inertial parameters. Therefore, we are naturally motivated to ask: can we construct two completely different inertial steps? In this subsection, we modify the second inertial step (3.1b) of Algorithm 1 by utilizing the information of the first inertial step (3.1a). Such a modification makes both inertial steps overlapped. Namely, we call the new algorithm Overlapped inertial proximal ExtraGradient Algorithm (OEGA). The specific iterative schemes of OEGA is presented in Algorithm 2.

Algorithm 2 Overlapped inertial proximal extragradient algorithm (OEGA).

```
Choose initial points $x_0$ and $x_1 \in \mathbb{R}^n$.
Select $\alpha_0, \beta_0, s, \eta$ satisfying Assumptions 1 and 3.
for $k = 1, 2, \cdots$ do
    $w_k = x_k + \alpha_k(x_k - x_{k-1}),$                              (4.1a)
    $v_k = w_k + \beta_k(w_k - w_{k-1}),$                              (4.1b)
    $y_k = \mathbf{prox}_{sg}\left(w_k - s\nabla f(v_k)\right),$       (4.1c)
    $x_{k+1} = \mathbf{prox}_{\eta g}\left(w_k - \eta\nabla f(y_k)\right).$  (4.1d)
end for
```

Remark 4.1. From the updating scheme (4.1a), we easily rewrite (4.1b) as

$$\begin{aligned} v_k &= w_k + \beta_k(w_k - w_{k-1}) \\ &= x_k + (\alpha_k + \beta_k + \alpha_k\beta_k)(x_k - x_{k-1}) - \alpha_{k-1}\beta_k(x_{k-1} - x_{k-2}). \end{aligned}$$

Therefore, the iterate v_k in (4.1b) contains more historical information than the one generated by (3.1b), which is possibly beneficial for improving the performance of Algorithm 2. We will show it in Section 5.

4.2. **Convergence analysis.** To begin this section, we first modify Assumption 2 as follows.

Assumption 3. *The inertial sequence $\{\alpha_k\}$ is non-decreasing and $\{\beta_k\}$ satisfy $0 < \alpha_k \leq \widehat{\alpha} < 1$ and $0 < \widetilde{\beta} \leq \beta_k < 1$. Moreover, for some $\bar{\delta} > 0$ and $\bar{\sigma} > 0$, we assume these parameters satisfying*

$$(\widehat{\alpha} + \bar{\delta})\left(\widehat{\alpha}(1+\widehat{\alpha}) + 4L\eta\left(\frac{Ls\widetilde{\beta}(\widehat{\alpha}+1)}{1-Ls}\right)^2 + 4L\eta\left(\frac{Ls\widetilde{\beta}\widehat{\alpha}}{1-Ls}\right)^2 + \upsilon\widehat{\alpha}\bar{\delta} + \bar{\sigma}\right) \leq \bar{\delta}\upsilon,$$

where υ is defined by (3.2).

Note that Algorithms 1 and 2 share similar iterative schemes. Below, we only present the main convergence theorem of Algorithm 2.

Theorem 4.2. *Suppose that Assumptions 1 and 3 hold. Then, the sequence $\{x_k\}$ generated by Algorithm 2 converges to a solution of* (1.1).

Proof. It follows from (3.15), (3.17) and (3.19) that

$$\begin{aligned} &\|x_{k+1} - x^*\|^2 - (1+\alpha_k)\|x_k - x^*\|^2 + \alpha_k\|x_{k-1} - x^*\|^2 \\ &\leq -\left(1 - 2L\eta\left(\frac{1}{1-Ls} - \frac{s}{\eta}\right)^2\right)\|w_k - x_{k+1}\|^2 \\ &\quad + \left(\alpha_k(1+\alpha_k) + 4L\eta\left(\frac{Ls\beta_k(\alpha_k+1)}{1-Ls}\right)^2\right)\|x_k - x_{k-1}\|^2 \\ &\quad + 4L\eta\left(\frac{Ls\beta_k\alpha_{k-1}}{1-Ls}\right)^2\|x_{k-1} - x_{k-2}\|^2 \\ &\leq -\left(1 - 2L\eta\left(1 - \frac{s}{\eta} + \frac{Ls}{1-Ls}\right)^2\right)\frac{\bar{\delta}}{\alpha_k + \bar{\delta}}\|x_{k+1} - x_k\|^2 \\ &\quad + \phi_k\|x_k - x_{k-1}\|^2 + \tau_{k-1}\|x_{k-1} - x_{k-2}\|^2, \end{aligned}$$

where

$$\phi_k = \alpha_k(1+\alpha_k) + 4L\eta\left(\frac{Ls\beta_k(\alpha_k+1)}{1-Ls}\right)^2 + \upsilon\alpha_k\bar{\delta}$$

and

$$\tau_{k-1} = 4L\eta\left(\frac{Ls\beta_k\alpha_{k-1}}{1-Ls}\right)^2.$$

Under Assumption 3, it follows from the boundedness of α_k and β_k that

$$\phi_k \le \widehat{\phi} := \widehat{\alpha}(1+\widehat{\alpha}) + 4L\eta\left(\frac{Ls\widetilde{\beta}(\widehat{\alpha}+1)}{1-Ls}\right)^2 + \upsilon\widehat{\alpha}\bar{\delta}$$

and

$$\tau_{k-1} \le \widehat{\tau} := 4L\eta\left(\frac{Ls\widetilde{\beta}\widehat{\alpha}}{1-Ls}\right)^2.$$

Below, we denote

$$\varphi_k := \|x_k - x^*\|^2 - \alpha_k\|x_{k-1} - x^*\|^2 + (\phi_k + \tau_k)\|x_k - x_{k-1}\|^2 + \tau_{k-1}\|x_{k-1} - x_{k-2}\|^2.$$

With the definition of φ_k, we have

$$\varphi_{k+1} - \varphi_k \le \left(\phi_{k+1} - \frac{\bar{\delta}\upsilon}{\alpha_k + \bar{\delta}} + \tau_{k+1}\right)\|x_{k+1} - x_k\|^2. \tag{4.2}$$

By the boundedness of α_k, ϕ_k, and τ_k, we claim that there exists some $\bar{\sigma} > 0$ such that

$$\phi_{k+1} - \frac{\bar{\delta}\upsilon}{\alpha_k + \bar{\delta}} + \tau_{k+1} \le -\bar{\sigma}. \tag{4.3}$$

Therefore, under Assumption 3, it is easy to get

$$(\alpha_k + \bar{\delta})(\phi_{k+1} + \tau_{k+1} + \bar{\sigma}) \le (\widehat{\alpha} + \bar{\delta})(\widehat{\phi} + \widehat{\tau} + \bar{\sigma}) \le \bar{\delta}\upsilon.$$

Consequently, it follows from (4.2) and (4.3) that

$$\varphi_{k+1} - \varphi_k \le -\bar{\sigma}\|x_{k+1} - x_k\|^2,$$

which implies that sequence $\{\varphi_k\}$ is non-increasing, and its boundness for $\{\alpha_k\}$ further delivers

$$\|x_k - x^*\|^2 - \widehat{\alpha}\|x_{k-1} - x^*\|^2 \le \varphi_k \le \varphi_1. \tag{4.4}$$

It immediately follows from (4.4) that

$$\|x_k - x^*\|^2 \le \widehat{\alpha}^k\|x_0 - x^*\|^2 + \varphi_1\sum_{n=0}^{k-1}\widehat{\alpha}^n \le \widehat{\alpha}^k\|x_0 - x^*\|^2 + \frac{\varphi_1}{1-\widehat{\alpha}}.$$

Thus, we obtain

$$\sum_{k=1}^{\infty}\|x_{k+1} - x_k\|^2 < +\infty$$

which show that

$$\|x_{k+1} - x_k\| \to 0.$$

Consequently, in some way analogous to (3.30)-(3.34), we have

$$\lim_{k\to\infty}\|w_k - x_{k+1}\| = \lim_{k\to\infty}\|w_k - v_k\| = \lim_{k\to\infty}\|x_k - y_k\| = 0.$$

The remainder proof is analogous to the proof Theorem 3.6. Hence, we omit it for the conciseness of this paper. The proof is complete. □

5. Numerical Experiments

In this section, we conduct the numerical performance of Algorithms 1 (DEGA) and 2 (OEGA) on two well-known statistical optimization problems: Dantzig selector and Lasso. To show our improvements, we compare DEGA and OEGA with some state-of-the-art gradient-like methods. Our code is written by Matlab 2021a, and all experiments are conducted on a 64-bit Windows PC equipped with Intel Core i5-12500h CPU@2.50GHz and 8GB RAM.

5.1. **The Dantzig selector problem.** We first consider the Dantzig selector problem introduced in [9]. Mathematically, it takes the form of

$$\min_{x\in\mathbb{R}^n}\left\{\|x\|_1 \mid \|D^{-1}K^\top(Kx-b)\|_\infty \le \delta\right\}, \tag{5.1}$$

where $K\in\mathbb{R}^{m\times n}$ is a design matrix, $D\in\mathbb{R}^{n\times n}$ is a diagonal matrix whose i-th diagonal term corresponds to the i-th column norm of K; $b\in\mathbb{R}^m$ is an observation vector; δ is a tuning parameter depending on the Gaussian noise standard deviation σ. To implement our algorithms for finding a solution of (5.1), we follow the way studied in [19] to reformulate (5.1) as the following form:

$$\min_x\left\{\|x\|_1+\frac{\varrho}{2}\left\|K^\top Kx-\mathbf{proj}_{\mathcal{Q}}(K^\top Kx)\right\|^2\right\}, \tag{5.2}$$

where $\varrho>0$ is a penalty parameter, $\mathcal{Q}=\{y \mid \|D^{-1}(y-K^\top b)\|_\infty\le\delta\}$ and $\mathbf{proj}_{\mathcal{Q}}(\cdot)$ represents the projection onto $\mathcal{Q}$. Obviously, (5.2) falls into the form of (1.1).

We consider the first case studied in [19], in which the design matrices have unit column norms. Specifically, we first randomly generate an $m\times n$ matrix with independent Gaussian components and then normalize each column with unit norm, which is the design matrix K in (5.2). Then we randomly generated a κ-sparse vector $x^\star\in\mathbb{R}^n$, which is generated by

$$x_i^\star := \begin{cases}\text{sign}(\xi_i)\times(1+|a_i|), & \text{if } i\in\mathcal{T},\\ 0, & \text{otherwise,}\end{cases}$$

where $\mathcal{T}$ is a random sample set with cardinality κ, $a_i\sim N(0,1)$, $\xi_i\sim U(-1,1)$, and $|\cdot|$ is the absolute value function. Finally, the observation vector b is generated via $b=Kx^\star+\epsilon$, where ϵ is a random Gaussian noise satisfying $\epsilon\sim N(0,\sigma^2 I)$. In our experiments, we will consider two scenarios on σ, i.e., $\sigma=0.05$ and $\sigma=0.10$. Moreover, we consider six cases on the problem setting, i.e., $(m,n,\kappa)=(720i,2560i,80i)$ with $i=1,2,\cdots,6$.

To support the improvements of our algorithm, we here only compare our DEGA with the extended extragradient (denoted by EEG, see (1.5)), the general inertial proximal gradient (GIPGM in short, see [39]), and the partially linearized ADMM (denoted by PLADM, see [19]). In our experiments, for the model parameters in (5.2), we take $\varrho=60$ and $\delta=\sqrt{2\log(n)}\sigma^{1.5}$. Then, for algorithmic parameters, we take the default settings for PLADM suggested in [19]. For the other three methods, we only consider constant settings for their parameters due to their similar iterative schemes. Concretely, we simply set $(\alpha_k,\beta_k,\lambda_k)=(0.92,0.84,\gamma)$ for GIPGM, $(s,\eta)=(\gamma,8.8\gamma)$ for EEG and $(\alpha_k,\beta_k,s,\eta)=(0.55,0.6,\gamma,6.9\gamma)$ for DEGA, where $\gamma=1/70$. Here, we should emphasize that we have tried our best to optimize all settings in our experiments. To ensure the fairness in comparisons, all algorithms start with the same initial points $\mathbf{0}=(0,\ldots,0)$, and stop at

$$\frac{\|x_k-x_{k-1}\|_\infty}{\max\{\|x_k\|_2,10\}}\le 10^{-5}.$$

To measure the accuracy of a solution, we follow the way used in [9]. More specifically, we let $\tilde{x}$ be an approximate solution obtained by algorithms, and let $\hat{x}$ be a post-processing solution of $\tilde{x}$ by the way suggested in [19]. Then, we define

$$\rho_{\text{orig}} := \frac{\sum_{i=1}^n(\tilde{x}_i-x_i^\star)^2}{\sum_{i=1}^n\min\{x_i^2,\sigma^2\}} \quad\text{and}\quad \rho_{\text{post}} := \frac{\sum_{i=1}^n(\hat{x}_i-x_i^\star)^2}{\sum_{i=1}^n\min\{x_i^2,\sigma^2\}}.$$

Clearly, the smaller values of ρ_{orig} and ρ_{post} mean better Dantzig selectors (i.e., solutions of (5.1)). In experiments, we generate ten groups of random data for each cases to demonstrate the stability of algorithm. The averaged numerical results, including number of iterations (Iter.), computing time in seconds (Time), ρ_{orig}, and ρ_{post}, are summarized in Table 1.

TABLE 1. Numerical results for finding Dantzig selectors

		$\sigma = 0.05$			$\sigma = 0.10$		
i	Method	Iter.	Time	ρ_{orig} (ρ_{post})	Iter.	Time	ρ_{orig} (ρ_{post})
1	DEGA	720.1	2.77	3.16 (0.49)	1374.7	5.10	3.69 (0.78)
	GIPGM	937.4	1.81	3.13 (0.46)	2012.7	3.74	3.65 (0.75)
	EEG	1138.9	4.39	3.23 (0.49)	1991.5	7.37	3.75 (0.79)
	PLADM	1565.3	4.56	1.90 (0.38)	2252.5	6.32	2.96 (0.54)
2	DEGA	717.5	16.67	3.44 (0.71)	1485.1	30.61	4.14 (0.63)
	GIPGM	953.5	10.84	3.39 (0.71)	2163.7	22.36	4.07 (0.62)
	EEG	1131.2	25.78	3.52 (0.71)	2133.2	43.90	4.25 (0.63)
	PLADM	1530.9	27.48	2.13 (0.47)	2150.9	33.34	3.36 (0.53)
3	DEGA	698.4	34.89	3.58 (0.55)	1505.7	73.42	4.39 (0.69)
	GIPGM	929.6	23.38	3.51 (0.55)	2206.4	53.60	4.33 (0.69)
	EEG	1068.7	53.40	3.75 (0.56)	2027.9	98.83	4.54 (0.70)
	PLADM	1425.7	53.34	2.28 (0.37)	2068.1	75.08	3.60 (0.57)
4	DEGA	691.7	63.48	3.53 (0.66)	1458.5	128.73	4.62 (0.79)
	GIPGM	916.2	42.07	3.48 (0.65)	2088.6	92.27	4.55 (0.77)
	EEG	1058.9	96.99	3.68 (0.68)	1972.8	173.66	4.83 (0.82)
	PLADM	1421.3	97.75	2.18 (0.45)	2035.2	134.78	3.77 (0.55)
5	DEGA	686.2	100.41	3.77 (0.64)	1359.3	195.73	4.78 (0.73)
	GIPGM	922.5	67.57	3.69 (0.63)	2087.4	150.52	4.67 (0.72)
	EEG	1046.8	153.22	3.98 (0.65)	1909.8	274.52	4.96 (0.75)
	PLADM	1322.3	145.50	2.38 (0.50)	1976.9	213.35	3.86 (0.58)
6	DEGA	672.2	138.99	3.76 (0.73)	1389.9	293.96	4.48 (0.82)
	GIPGM	916.8	94.88	3.66 (0.71)	1978.9	208.73	4.41 (0.81)
	EEG	1041.8	215.36	3.95 (0.76)	1886.7	397.26	4.67 (0.86)
	PLADM	1286.8	199.63	2.43 (0.46)	2028.2	321.52	3.61 (0.64)

It can be seen from Table 1 that the proposed DEGA runs the fastest in terms of taking the least iteration for both scenarios of $\sigma = 0.05$ or $\sigma = 0.1$. When comparing DEGA and GIPGM, although DEGA takes fewer iterations than GIPGM, the former taking more computing time than the latter. The main reason is due to the one more evaluation on the gradient, which includes high-dimensional matrix products. However, when comparing our DEGA with the EEG, the former performs much better than the latter for large-scale cases, which sufficiently verifies that our inertial strategy is practical for improving the performance of the extragradient method (1.5). Note that PLADM is a powerful splitting method tailored for some structured convex optimization problems. The results in Table 1 demonstrate that our DEGA is competitive when comparing it with PLADM.

Due to the randomness of the generated data, we can only see the averaged performance of all compared algorithms. Therefore, to investigate the stability of the four algorithms, we show their mean iterations and corresponding standard errors for each case with setting $\sigma = 0.05$ in Fig. 1. Comparatively, our DEGA runs a little more stable than the other three algorithms, which further supports idea of this paper.

5.2. **The Lasso problem.** In this subsection, we further consider a well-known Lasso problem [14], which is a fundamental problem in statistical learning and plays an important role in compressive

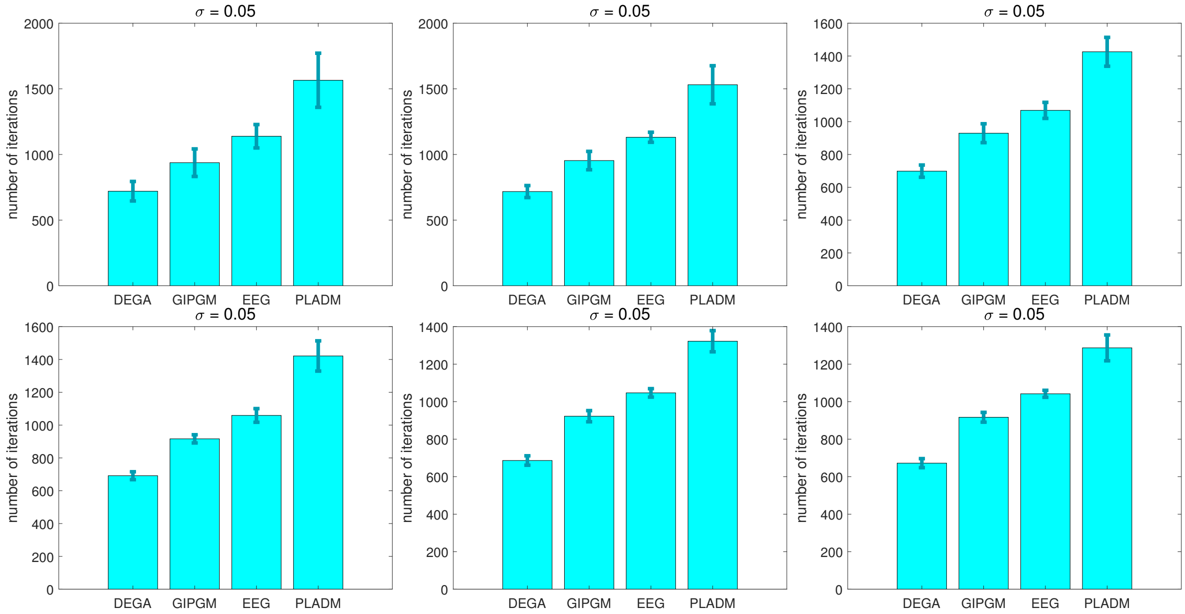

FIGURE 1. Mean iterations and its standard error. From left to right, top to bottom: The figures correspond to $i = 1, 2, \ldots, 6$, respectively, with setting $\sigma = 0.05$

sensing and machine learning. The Lasso model reads as

$$\min_{x} \left\{ F(x) := \varrho \|x\|_1 + \frac{1}{2}\|Ax - b\|^2 \right\}, \tag{5.3}$$

where $A \in \mathbb{R}^{m\times n}$ is a design matrix, $b \in \mathbb{R}^m$ is an observation vector, and ϱ is a tuning parameter. Obviously, (5.3) falls into the standard form of (1.1). Therefore, the proposed two new algorithms are applicable to (5.3).

Below, we first introduce the way of generating problem data. We randomly generate matrix $K \in \mathbb{R}^{m\times n}$ with independent Gaussian entries, and normalize it by $A = K/\|K\|$. Then, we generate a random κ-sparse vector $x^\star$ by the following MATLAB script:

$x^\star$=zeros(n,1); xs=randn(k,1); P=randi(n,k,1); $x^\star$(P)=xs,

Finally, the observation vector $b \in \mathbb{R}^n$ is set as $b = Ax^\star$. Therefore, it is clear that $x^\star$ is the true solution of (5.3).

Considering that the proposed OEGA is not compared in Section 5.1, in this part, we compare it with EEG, GIPGM, and DEGA. In our experiments, we conduct eight cases of the problem size, i.e, $(m, n, \kappa) = (256i, 1024i, 40i)$ with $i = 1, 2, \cdots, 8$. For the model parameter in (5.3), we take $\varrho = 5 \times 10^{-3}$. Like the settings in Section 5.1, for the algorithmic parameters, we set $(s, \eta) = (0.78, 6.25)$ for EEG; $(\alpha_k, \beta_k, \lambda_k) = (0.75, 0.6, 1.0)$ for GIPGM; $(\alpha_k, \beta_k, s, \eta) = (0.46, 0.475, 1.0, 6.2)$ for DEGA, $(\alpha_k, \beta_k, s, \eta) = (0.41, 0.13, 1.05, 8.1)$ for OEGA. To ensure the fairness in comparison, we take all starting points as zeros and the following stopping criterion:

$$\max\{\|x_k - x_{k-1}\|, |F(x_k) - F(x_{k-1})|\} \leq 10^{-10}.$$

Due to the randomness of data, we report the averaged performance of the four algorithms for each case. All results are summarized in Table 2.

It can be seen from Table 2 that our DEGA and OEGA outperform EEG and GIPGM in terms of iterations and computing time. Unlike the results reported in Section 5.1, both DEGA and OEGA take less computing time than the GIPGM, although our algorithms require one more evaluation on gradient. The main reason is that the complexity of evaluating gradient of (5.3) is much lower than the one in

TABLE 2. Numerical results for solving Lasso

	EEG	GIPGM	DEGA	OEGA
i	Iter / Time	Iter / Time	Iter / Time	Iter / Time
1	155.1 / 0.035	214.2 / 0.034	76.0 / 0.017	70.7 / 0.016
2	160.1 / 0.336	211.8 / 0.315	75.2 / 0.164	66.6 / 0.141
3	161.3 / 1.039	217.5 / 0.984	76.7 / 0.496	65.4 / 0.416
4	161.0 / 2.247	218.3 / 2.138	75.9 / 1.060	64.6 / 0.908
5	163.8 / 3.742	215.4 / 3.470	75.6 / 1.732	64.8 / 1.488
6	172.8 / 5.837	222.6 / 5.320	77.9 / 2.636	67.0 / 2.289
7	164.3 / 7.576	217.9 / 7.063	76.2 / 3.507	64.7 / 2.993
8	164.0 / 10.112	216.4 / 9.391	75.6 / 4.640	64.5 / 3.964

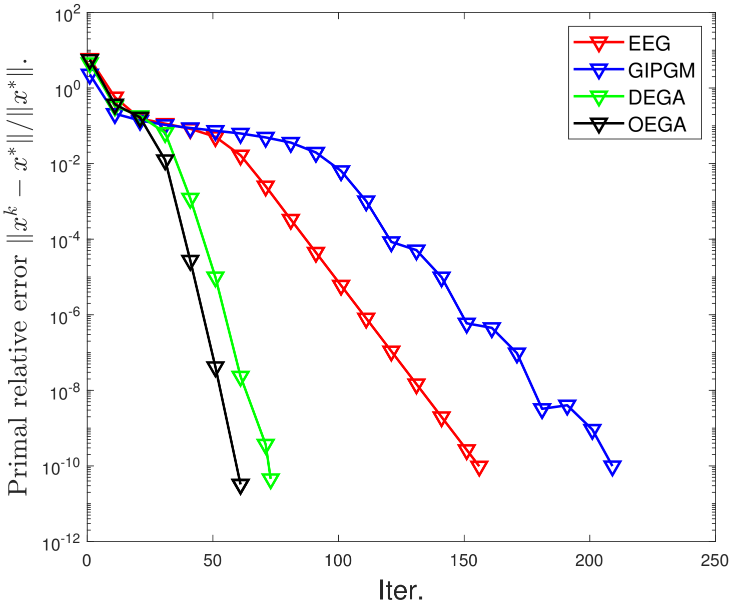

FIGURE 2. Convergence behaviors of the four algorithms for solving (5.3) with $(m, n, \kappa) = (512, 2048, 80)$

(5.1). In this case, the much fewer iterations can save some computing time, so that our algorithms (i.e., DEGA and OEGA) runs faster than GIPGM. These results in Table 2 show that our inertial-type extragradient methods are reliably practical for some real-world problems.

In Fig. 2, we plot the convergence curves of the four algorithms for solving (5.3) with $(m, n, \kappa) = (512, 2048, 80)$. As studied in [26], the original extragradient method has a sublinear convergence rate. Although we cannot establish the sublinear convergence rates for our algorithms, we can see from Fig. 2 that they possibly have the same convergence rate as the original extragradient method. In the future, we will consider this issue.

To make a complement illustration of Fig.3, we here show the box plots with respect to iterations of the four algorithms solving (5.3), where we only show the first four cases for the problem size, i.e., $(m, n, \kappa) = (256i, 1024i, 40i)$ with $i = 1, 2, 3, 4$. Clearly, all results show that our algorithms work more stable than the other two algorithms. Moreover, all reported results on solving (5.3) indicate that the OEGA works a little better than the DEGA, which further support that more historical information in (4.1b) is helpful in algorithmic acceleration.

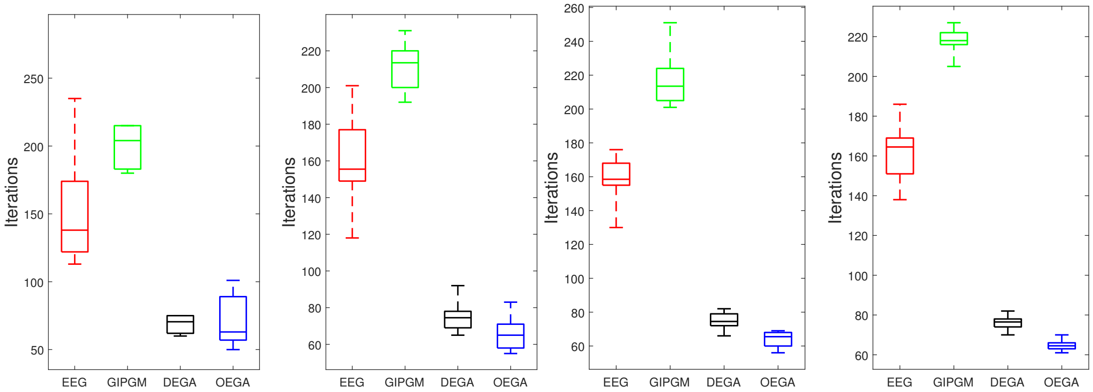

FIGURE 3. Box plots with respect to iterations of the four algorithms solving (5.3) with the first four cases, i.e., $(m, n, \kappa) = (256i, 1024i, 40i)$ with $i = 1, 2, 3, 4$. From left to right: they correspond to $i = 1, 2, 3, 4$, respectively

6. CONCLUSIONS

In this paper, we introduced two inertial proximal extragradient algorithms for solving a class of nonsmooth convex composite optimization problems. Both new algorithms have two inertial steps, which make them flexible in algorithmic implementation. Moreover, the second new algorithm inherits more historical information, which is possibly helpful for further improvements. Under some conditions, we proved that both algorithms are globally convergent. Comparatively, our algorithms perform well and stable in our experiments. In the future, we will consider some more general cases where the objective function is nonsmooth and nonconvex.

STATEMENTS AND DECLARATIONS

Data Availability: Data sharing is not applicable to this article as no datasets were generated. The MATLAB code should be directed to the authors.
Conflict of interest: The authors declare that they have no competing interests.

ACKNOWLEDGMENTS

H. He was supported in part by Natural Science Foundation of Zhejiang Province (No. LZ24A010001), Ningbo Natural Science Foundation (No. 2023J014), and National Natural Science Foundation of China (No. 12371303).

REFERENCES

[1] F. Alvarez. Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space. *SIAM Journal on Optimization*, 14:773–782, 2004.
[2] H. Attouch, J. Peypouquet, and P. Redont. A dynamical approach to an inertial forward-backward algorithm for convex minimization. *SIAM Journal on Optimization*, 24(1):232–256, 2014.
[3] H. Bauschke and P. Combettes. *Correction to: convex analysis and monotone operator theory in Hilbert spaces*. Springer International Publishing, New York, 2017.
[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
[5] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. In D. P. Palomar and Y. C. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 33–88, New York, 2010. Cambridge University Press.
[6] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, 3rd edition, 2016.
[7] R. Boţ, E. Csetnek, and S. László. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4:3-25, 2016.
[8] C. A. Bouman. *Foundations of Computational Imaging: A Model-Based Approach*. SIAM, Philadelphia, 2022.

[9] E. Candés and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35:2313–2351, 2007.

[10] Y. Censor, A. Gibali, and S. Reich. Extensions of Korpelevich's extragradient method for the variational inequality problem in Euclidean space. *Optimization*, 61(9):1119–1132, 2012.

[11] C. Chen, R. Chan, S. Ma, and J. Yang. Inertial proximal ADMM for linearly constrained separable convex optimization. *SIAM Journal on Imaging Sciences*, 8(4):2239–2267, 2015.

[12] G. Chen and R. Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, 1997.

[13] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[14] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–451, 2004.

[15] X. Gao, X. Cai, and D. Han. A Gaus-Seidel type inertial proximal alternating linearized minimization for a class of nonconvex optimization problems. *Journal of Global Optimization*, 76:863–887, 2020.

[16] A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70:709–710, 1964.

[17] B. He and L. Liao. Improvements of some projection methods for monotone nonlinear variational inequalities. *Journal of Optimization Theory and Applications*, 112:111–128, 2002.

[18] B. He, X. Yuan, and J. Zhang. Comparison of two kinds of prediction-correction methods for monotone variational inequalities. *Computational Optimization and Applications.*, 27:247–267, 2004.

[19] H. He and H.-K. Xu. Splitting methods for split feasibility problems with application to Dantzig selectors. *Inverse Problems*, 33:Article ID 055003, 2017.

[20] P. Johnstone and P. Moulin. Local and global convergence of a general inertial proximal splitting scheme for minimizing composite functions. *Computational Optimization and Applications.*, 67(2):259–292, 2017.

[21] A. Juditsky and A. Nemirovski. *Statistical Inference via Convex Optimization*. Princeton University Press, New Jersey, 2020.

[22] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[23] E. Levitin and B. Polyak. Constrained minimization problems. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 1966.

[24] J. Moreau. Proximité et dualité dans un espace Hilbertien. *Bulletin de la Société Mathématique de France*, 95:153–171, 1965.

[25] A. Moudafi and M. Oliny. Convergence of a splitting inertial proximal method for monotone operators. *Journal of Computational and Applied Mathematics*, 155(2):447–454, 2003.

[26] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequality with Lipschitz continuous monotone operators and smooth convex-concave saddle points problems. *SIAM Journal on Optimization*, 15:229–251, 2004.

[27] Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269:543–547, 1983.

[28] T. Nguyen, E. Pauwels, E. Richard, and B. Suter. Extragradient method in optimization: Convergence and complexity. *Journal of Optimization Theory and Applications*, 176:137–162, 2018.

[29] P. Ochs, T. Brox, and T. Pock. iPiasco: Inertial proximal algorithm for strongly convex optimization. *Journal of Mathematical Imaging and Vision*, 53:171–181, 2015.

[30] P. Ochs, Y. Chen, T. Brox, and T. Pock. Ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[31] Z. Opial. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bulletin of the American Mathematical Society*, 73:591–597, 1967.

[32] D. Palomar and Y. Eldar. *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, England, 2010.

[33] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1:123–231, 2013.

[34] Y. Qu, H. He, and D. Han. A partially inertial customized Douglas-Rachford splitting method for a class of structured optimization problems. *Journal of Scientific Computing*, 98:Artical ID. 9, 2024.

[35] S. Sra, S. Nowozin, and S. Wright. *Optimization for Machine Learning*. MIT Press, 2012.

[36] D. Thong, N. Vinh, and Y. Cho. Accelerated subgradient extragradient methods for variational inequality problems. *Journal of Scientific Computing*, 80:1438–1462, 2019.

[37] Q. Tran-Dinh. Sublinear convergence rates of extragradient-type methods: A survey on classical and recent developments. arXiv:2303.17192v1, 2023.

[38] K. Wang and H. He. A double extrapolation primal-dual algorithm for saddle point problems. *Journal of Scientific Computing*, 85(3):1–30, 2020.

[39] Z. Wu and M. Li. General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems. *Computational Optimization and Applications.*, 73:129-158, 2019.

[40] Z. Xie, G. Cai, X. Li, and Q. Dong. Strong convergence of the modified inertial extragradient method with line-search process for solving variational inequality problems in hilbert spaces. *Journal of Scientific Computing*, 88:Article ID 50, 2021.

[41] Y. Yao, O. Iyiola, and Y. Shehu. Subgradient extragradient method with double inertial steps for variational inequalities. *Journal of Scientific Computing*, 90:Article ID 71, 2022.

[42] J. Zhao, Q. Dong, M. Rassias, and F. Wang. Two-step inertial Bregman alternating minimization algorithm for nonconvex and nonsmooth problems. *Journal of Global Optimization*, 84:941–966, 2022.