Adaptive and Universal Algorithms for Variational Inequalities with Optimal Convergence

Alina Ene[*1], **Huy L. Nguyen**[*2]

[1]Boston University,
[2]Northeastern University
aene@bu.edu, hu.nguyen@northeastern.edu

Abstract

We develop new adaptive algorithms for variational inequalities with monotone operators, which capture many problems of interest, notably convex optimization and convex-concave saddle point problems. Our algorithms automatically adapt to unknown problem parameters such as the smoothness and the norm of the operator, and the variance of the stochastic evaluation oracle. We show that our algorithms are universal and simultaneously achieve the optimal convergence rates in the non-smooth, smooth, and stochastic settings. The convergence guarantees of our algorithms improve over existing adaptive methods and match the optimal non-adaptive algorithms. Additionally, prior works require that the optimization domain is bounded. In this work, we remove this restriction and give algorithms for unbounded domains that are adaptive and universal. Our general proof techniques can be used for many variants of the algorithm using one or two operator evaluations per iteration. The classical methods based on the ExtraGradient/MirrorProx algorithm require two operator evaluations per iteration, which is the dominant factor in the running time in many settings.

1 Introduction

Variational inequalities with monotone operators are a general framework for solving problems with convex structure including convex minimization, convex-concave saddle point problems, and finding convex Nash equilibrium (Nemirovski 2004; Juditsky, Nemirovski, and Tauvel 2011). Given a convex domain $\mathcal{X} \subseteq \mathbb{R}^d$ and a monotone mapping $F : \mathcal{X} \to \mathbb{R}^d$,

$$\langle F(x) - F(y), x - y \rangle \geq 0 \; \forall x, y \in \mathcal{X}$$

we are interested in finding an approximation to a solution x^* such that[1]

$$\langle F(x^*), x^* - x \rangle \leq 0 \; \forall x \in \mathcal{X}$$

More recently, algorithms developed in this framework are also applied to non-convex problems including optimizing generative adversarial networks (GANs) (Daskalakis et al. 2018; Yadav et al. 2018; Chavdarova et al. 2019; Gidel et al. 2019; Mertikopoulos et al. 2019). In this context, due to the large scale of the problems, several important issues are brought to the fore. First, the algorithms typically require careful settings of the step sizes based on the parameters of the problems such as smoothness, especially for high dimensional problems where the smoothness varies for different coordinates. Second, classical methods based on the extra gradient algorithm (Korpelevich 1976) or the more general mirror prox algorithm (Nemirovski 2004) requires two gradient computations per iteration, which is the dominant factor in the running time, making them twice as slow as typical gradient descent methods. To rectify the first issue, several works have been developed to design adaptive algorithms that automatically adapt to the smoothness of the problem (Bach and Levy 2019; Ene, Nguyen, and Vladu 2021). These works build upon the impressive body of works that brought about adaptive algorithms for convex optimization methods (see e.g. (McMahan and Streeter 2010; Duchi, Hazan, and Singer 2011; Kingma and Ba 2014)). A different line of work focused on reducing the number of gradient computation to one per iteration (Popov 1980; Gidel et al. 2019; Hsieh et al. 2019; Chambolle and Pock 2011; Malitsky 2015; Cui and Shanbhag 2016; Daskalakis et al. 2018; Mokhtari, Ozdaglar, and Pattathil 2020). It is worth noting that in practice, especially in the context of training GANs, these methods are almost always used in a heuristic fashion along with adaptive techniques such as Adam (Kingma and Ba 2014).

In this work, we develop new algorithms achieving the best of both worlds: our algorithms automatically adapt to the smoothness of the problem and require only one gradient computation per iteration. We include two variants of the core algorithm, one variant adapts to a single shared smoothness parameter for all coordinates and the other variant adapts simultaneously to different smoothness parameters for different coordinates. Our algorithms can be viewed as adaptive versions of the past extra-gradient method developed by Popov (1980) and further analyzed by many subsequent works including most recently (Gidel et al. 2019; Hsieh et al. 2019). Our algorithms are universal: they work simultaneously for non-smooth functions, smooth functions, and with stochastic oracle access. In each of these settings,

[*]These authors contributed equally.

[1]Such a solution is called a strong solution. Following previous work, we design algorithms that converge to a weak solution. If F is monotone and continuous, a weak solution is a strong solution and vice-versa. We defer the formal definitions to Section 2.

the algorithm adapting to the scalar smoothness parameter achieves the same convergence guarantees as the best-known algorithms using the smoothness parameter in their step sizes. In contrast, previous adaptive algorithms (Bach and Levy 2019; Ene, Nguyen, and Vladu 2021) lose logarithmic factors compared with non-adaptive algorithms and use twice as many operator evaluations. Furthermore, our algorithm for scalar smoothness allows for arbitrary initialization of the normalization factor, which is in line with the practice of initializing it to a small constant such as 10^{-10}. In contrast, previous works need the initial value to be at least the maximum operator value or the radius of the domain. Our analysis framework is general and versatile, and it allows us to analyze several variants of our algorithms, including algorithms based on the extra-gradient method (Korpelevich 1976) and algorithms that are suitable for unbouded optimization domains. A detailed comparison of the convergence rates is described in Table 1. We provide a discussion of the algorithmic and technical contributions in Section 3 and the full version (Ene and Nguyen 2021). We note that the convergence guarantees obtained by our scalar algorithm are optimal in all settings (non-smooth, smooth, and stochastic), as they match known lower bounds for convex optimization and saddle-point problems (Nemirovsky and Yudin 1983; Nemirovsky 1992; Ouyang and Xu 2021). Moreover, all of our algorithms automatically adapt to unknown problem parameters such as the smoothness and the norm of the operator, and the variance of the stochastic evaluation oracle.

2 Preliminaries

Variational inequalities: In this paper, we consider the problem of finding strong solutions to variational inequalities with monotone operators. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a non-empty closed convex set ($\mathcal{X}$ may be unbounded). Let $F\colon \mathcal{X} \to \mathbb{R}^d$ be a map. The variational inequality problem is to find a solution $x^* \in \mathcal{X}$ satisfying

$$\langle F(x^*), x^* - x\rangle \le 0 \quad \forall x \in \mathcal{X} \tag{1}$$

A solution x^* satisfying the above condition is often called a *strong solution* to the variational inequality.

The operator F is *monotone* if it satisfies

$$\langle F(x) - F(y), x - y\rangle \ge 0 \quad \forall x, y \in \mathcal{X} \tag{2}$$

A related notion is a *weak solution*, i.e., a point $x^* \in \mathcal{X}$ satisfying

$$\langle F(x), x^* - x\rangle \le 0 \quad \forall x \in \mathcal{X} \tag{3}$$

If F is monotone and continuous, a weak solution is a strong solution and vice-versa.

Let $\|\cdot\|$ be a norm and let $\|\cdot\|_*$ be its dual norm. The operator F is β-*smooth* with respect to the norm $\|\cdot\|$ if it satisfies

$$\|F(x) - F(y)\|_* \le \beta \|x - y\| \tag{4}$$

The operator F is β-*cocoercive* with respect to the norm $\|\cdot\|$ if it satisfies

$$\langle F(x) - F(y), x - y\rangle \ge \frac{1}{\beta}\|F(x) - F(y)\|_*^2 \quad \forall x, y \in \mathcal{X} \tag{5}$$

Using Holder's inequality, we can readily verify that, if F is β-cocoercive, then it is monotone and β-smooth.

Special cases: Two well-known special cases of the variational inequality problem with monotone operators are convex minimization and convex-concave saddle point problems.

In convex minimization, we are given a convex function $f\colon \mathcal{X} \to \mathbb{R}^d$ and the goal is to find a solution $x^* \in \arg\min_{x\in\mathcal{X}} f(x)$. The operator is the gradient of f, i.e., $F = \nabla f$ (if f is not differentiable, the operator is a subgradient of f). The monotonicity condition (2) is equivalent to f being convex. A strong solution is a point x^* that satisfies the first-order optimality condtion and thus it is a global minimizer of f. The smoothness condition (4) coincides with the usual smoothness condition from convex optimization. If f is convex and β-smooth, then $F = \nabla f$ is β-cocoercive (see, e.g., Theorem 2.1.5 in the textbook (Nesterov 2013)).

In convex-concave saddle point problems, we are given a function $f\colon \mathcal{U} \times \mathcal{V} \to \mathbb{R}^d$ such that $f(u, v)$ is convex in u and concave in v, and the goal is to solve $\min_{u\in\mathcal{U}} \max_{v\in\mathcal{V}} f(u, v)$. The operator is $F = (\nabla_u f, -\nabla_v f)$. A strong solution is a point (u^*, v^*) that is a global saddle point, i.e.,

$$f(u^*, v) \le f(u^*, v^*) \le f(u, v^*) \quad \forall (u, v) \in \mathcal{U} \times \mathcal{V}$$

Error function: Following previous work (Nemirovski 2004; Nesterov 2007), we analyze convergence via the error (or merit) function. Following (Nesterov 2007), we choose an arbitrary point $x_0 \in \mathcal{X}$. For any fixed positive value D, we define

$$\mathrm{Err}_D(x) = \sup_{y\in\mathcal{X}} \{\langle F(y), x - y\rangle : \|y - x_0\| \le D\} \tag{6}$$

If $\mathcal{X}$ is a bounded domain, we define

$$\mathrm{Err}(x) = \sup_{y\in\mathcal{X}} \langle F(y), x - y\rangle \tag{7}$$

The following lemma, shown in (Nesterov 2007), justifies the use of the error function to analyze convergence.

Lemma 2.1. (Nesterov 2007) *Let D be any fixed positive value. The function Err_D is well-defined and convex on $\mathbb{R}^d$. For any $x \in \mathcal{X}$ such that $\|x - x_0\| \le D$, we have $\mathrm{Err}_D(x) \ge 0$. If x^* is a weak solution and $\|x^* - x_0\| \le D$, then $\mathrm{Err}_D(x^*) = 0$. Moreover, if $\mathrm{Err}_D(x) = 0$ for some $x \in \mathcal{X}$ with $\|x - x_0\| < D$, then x is a weak solution.*

We will use the following inequalities that were shown in previous work.

Lemma 2.2. (Duchi, Hazan, and Singer 2011; McMahan and Streeter 2010) *Let $a_1, \ldots, a_T$ be non-negative scalars. We have*

$$\sqrt{\sum_{t=1}^T a_t} \le \sum_{t=1}^T \frac{a_t}{\sqrt{\sum_{s=1}^t a_s}} \le 2\sqrt{\sum_{t=1}^T a_t}$$

Lemma 2.3. (Bach and Levy 2019) *Let $a_1, \ldots, a_T \in [0, a]$ be non-negative scalars that are at most a. Let $a_0 \ge 0$. We*

Non-smooth	Smooth
$O\left(\frac{R(G+\sigma)}{\sqrt{T}}\right)$	$O\left(\frac{\beta R^2}{T}+\frac{R\sigma}{\sqrt{T}}\right)$
Theorem 3.1. **Scalar** step sizes, **bounded** domain, **1 evaluation** per iteration.	
$O\left(\frac{\widehat{G}R\sqrt{\ln T}}{\sqrt{T}}\right)$	$O\left(\frac{\widehat{G}R+\beta R^2\left(1+\ln\left(\beta R/\widehat{G}\right)\right)+G^2}{T}+\frac{R\sigma\sqrt{\ln T}}{\sqrt{T}}\right)$
Bach and Levy (2019). **Scalar** step sizes, **bounded** domain, **2 evaluations** per iteration.	
$O\left(\frac{\|x_0-x^*\|^2+G^2}{T}+\frac{\|x_0-x^*\|(G+\sigma)}{\sqrt{T}}\right)$	$O\left(\frac{\beta\|x_0-x^*\|^2+\|x_0-x^*\|G+G^2}{T}+\frac{\|x_0-x^*\|\sigma}{\sqrt{T}}\right)$
Theorem 3.2. **Scalar** step sizes, **arbitrary** domain, **1 evaluation** per iteration.	
$O\left(\frac{\|x_0-x^*\|^2+G^3+G\ln\left(1+G^2T\right)}{\sqrt{T}}\right)$	$O\left(\frac{\left(\beta\|x_0-x^*\|^2+\beta^4+\beta^2\left\|F(x_{t_0+1/2})-F(x_{t_0})\right\|^2\right)^{3/2}}{T}\right)$ t_0 is the last iteration t such that $\eta_{t_0}\geq\frac{c}{\beta}$ for constant c
Antonakopoulos et al. (2021). **Seterministic** ($\sigma=0$), **scalar** step sizes, **arbitrary** domain, **2 evaluations** per iteration.	
$O\left(\frac{\|x_0-x^*\|^2+G^2}{T}+\frac{\|x_0-x^*\|G}{\sqrt{T}}\right)$	$O\left(\frac{\beta^2\|x_0-x^*\|^2+\|F(x_\tau)-F(x_{\tau-1})\|^2+\|F(x_{\tau-1})-F(x_{\tau-2})\|^2}{T}\right)$ τ is the last iteration t such that $\gamma_{t-2}\leq c\beta$ for constant c
Full version. **Seterministic** ($\sigma=0$), **scalar** step sizes, **arbitrary** domain, **1 evaluation** per iteration.	
$O\left(\frac{dR_\infty^2}{T}+\frac{\sqrt{d}R_\infty G+\left(\sqrt{d}R_\infty+R\right)\sigma}{\sqrt{T}}\right)$	$O\left(\frac{dR_\infty^2\beta^2}{T}+\frac{\left(\sqrt{d}R_\infty+R\right)\sigma}{\sqrt{T}}\right)$
Full version. **Vector** step sizes, **bounded** domain, **1 evaluation** per iteration.	
$O\left(\frac{dR_\infty^2}{T}+\frac{\sqrt{d}R_\infty G\left(\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}\right)+R\sigma}{\sqrt{T}}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T}+\frac{R\sigma}{\sqrt{T}}\right)$
Full version. **Vector** step sizes, **bounded** domain, **1 evaluation** per iteration.	
$O\left(\frac{dR_\infty^2}{T}+\frac{\sqrt{d}R_\infty\left(G\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}+\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}\right)}{\sqrt{T}}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T}+\frac{\sqrt{d}R_\infty\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}\right)$
Ene, Nguyen, and Vladu (2021). **Vector** step sizes, **bounded** domain, **2 evaluations** per iteration.	

Table 1: Comparison of adaptive algorithms for variational inequalities. R, R_∞ are the ℓ_2 and ℓ_∞ diameter of the domain. G is an upper bound on the ℓ_2-norm of $F(\cdot)$. σ^2 is the variance of the stochastic oracle for $F(\cdot)$ (for deterministic setting, set $\sigma=0$). d is the dimensions of the domain. In the smooth setting, F is smooth with respect to a norm $\|\cdot\|_{\mathbf{B}}$, where $\mathbf{B}=\operatorname{diag}(\beta_1,\ldots,\beta_d)$ is a diagonal matrix with $\beta_1,\ldots,\beta_d>0$; we let $\beta=\max_i\beta_i$. The scalar algorithms set a single step size for all coordinates, whereas the vector algorithms set a per-coordinate step size. The stated bounds are obtained by setting $\gamma_0=0$ in Theorem 3.1 and $\gamma_0=1$ In Theorem 3.2. The analysis of Bach and Levy (2019) requires the stochastic gradients to be bounded almost surely by a parameter $\widehat{G}$, which is stronger than the variance assumption we use in this paper. Additionally, the algorithm of Bach and Levy (2019) requires an estimate for $\widehat{G}$ in order to step size.

have

$$\sqrt{a_0+\sum_{t=1}^{T-1}a_t}-\sqrt{a_0}\leq\sum_{t=1}^{T}\frac{a_t}{\sqrt{a_0+\sum_{s=1}^{t-1}a_s}}$$
$$\leq\frac{2a}{\sqrt{a_0}}+3\sqrt{a}+3\sqrt{a_0+\sum_{t=1}^{T-1}a_t}$$

We will also make use of the following facts from Fenchel duality. Let $\phi\colon\mathcal{X}\to\mathbb{R}$ be a differentiable convex function. The function ϕ is β-smooth with respect to the norm $\|\cdot\|$ if

$$\phi(y)\leq\phi(x)+\langle\nabla\phi(x),y-x\rangle+\frac{\beta}{2}\|y-x\|^2\quad\forall x,y\in\mathcal{X}$$

The function ϕ is α-strongly convex with respect to $\|\cdot\|$ if

$$\phi(y)\geq\phi(x)+\langle\nabla\phi(x),y-x\rangle+\frac{\alpha}{2}\|y-x\|^2\quad\forall x,y\in\mathcal{X}$$

The Fenchel conjugate of ϕ is the function $\phi^*\colon\mathcal{X}\to\mathbb{R}$ with

$$\phi^*(z)=\max_{x\in\mathcal{X}}\{\langle x,z\rangle-\phi(x)\}\quad\forall z\in\mathcal{X}$$

Lemma 2.4. ((Shalev-Shwartz et al. 2011), Lemma 2.19) *Let $\phi\colon\mathcal{X}\to\mathbb{R}$ be a closed convex function. The function ϕ is α-strongly convex with respect to a norm $\|\cdot\|$ if and only if ϕ^* is $\frac{1}{\alpha}$-smooth with respect to the dual norm $\|\cdot\|_*$.*

Lemma 2.5. (Danskin's theorem, (Bertsekas, Nedic, and Ozdaglar 2003), Proposition 4.5.1) *Let $\phi\colon\mathcal{X}\to\mathbb{R}$ be a strongly convex function. For all $v\in\mathcal{X}$, we have*

$$\nabla\phi^*(v)=\arg\min_{u\in\mathcal{X}}\{\phi(u)-\langle u,v\rangle\}$$

Additional notation: Throughout the paper, the norm $\|\cdot\|$ without a subscript denotes the standard ℓ_2-norm. We also use the Mahalanobis norm $\|x\|_{\mathbf{A}}:=\sqrt{x^\top\mathbf{A}x}$, where

Algorithm 1: AdaPEG algorithm for bounded domains $\mathcal{X}$.

Let $x_0 = z_0 \in \mathcal{X}$, $\gamma_0 \geq 0$, $\eta > 0$.
For $t = 1, \ldots, T$, update:

$$x_t = \arg\min_{u \in \mathcal{X}} \left\{ \left\langle \widehat{F(x_{t-1})}, u \right\rangle + \frac{1}{2}\gamma_{t-1} \|u - z_{t-1}\|^2 \right\}$$

$$z_t = \arg\min_{u \in \mathcal{X}} \left\{ \left\langle \widehat{F(x_t)}, u \right\rangle + \frac{1}{2}\gamma_{t-1} \|u - z_{t-1}\|^2 + \frac{1}{2}\left(\gamma_t - \gamma_{t-1}\right) \|u - x_t\|^2 \right\}$$

$$\gamma_t = \frac{1}{\eta}\sqrt{\eta^2\gamma_0^2 + \sum_{s=1}^{t} \left\| \widehat{F(x_s)} - \widehat{F(x_{s-1})} \right\|^2}$$

Return $\overline{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$.

Algorithm 2: AdaPEG algorithm for unbounded domains $\mathcal{X}$.

Let $x_0 = z_0 \in \mathcal{X}$, $\gamma_0 \geq 0$,$\gamma_{-1} = 0$, $\eta > 0$.
For $t = 1, \ldots, T$, update:

$$x_t = \arg\min_{u \in \mathcal{X}} \left\{ \left\langle \widehat{F(x_{t-1})}, u \right\rangle + \frac{1}{2}\gamma_{t-2} \|u - z_{t-1}\|^2 + \frac{1}{2}\left(\gamma_{t-1} - \gamma_{t-2}\right) \|u - x_0\|^2 \right\}$$

$$z_t = \arg\min_{u \in \mathcal{X}} \left\{ \left\langle \widehat{F(x_t)}, u \right\rangle + \frac{1}{2}\gamma_{t-2} \|u - z_{t-1}\|^2 + \frac{1}{2}\left(\gamma_{t-1} - \gamma_{t-2}\right) \|u - x_0\|^2 \right\}$$

$$\gamma_t = \frac{1}{\eta}\sqrt{\eta^2\gamma_0^2 + \sum_{s=1}^{t} \left\| \widehat{F(x_s)} - \widehat{F(x_{s-1})} \right\|^2}$$

Return $\overline{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$.

$\mathbf{A} \in \mathbb{R}^{d \times d}$ is a positive definite matrix. The dual norm of $\|\cdot\|_{\mathbf{A}}$ is $\|\cdot\|_{\mathbf{A}^{-1}}$. For a diagonal matrix $\mathbf{D} \in \mathbb{R}^{d \times d}$, we let $\mathbf{D}_i$ denote the i-th diagonal entry of $\mathbf{D}$ and we let $\mathrm{Tr}(\mathbf{D}) = \sum_{i=1}^{d} \mathbf{D}_i$ denote the trace of $\mathbf{D}$. For bounded domains $\mathcal{X}$, we let R and R_∞ denote the ℓ_2 and ℓ_∞ diameter of $\mathcal{X}$: $R = \max_{x,y \in \mathcal{X}} \|x - y\|$, $R_\infty = \max_{x,y \in \mathcal{X}} \|x - y\|_\infty$. We let $G = \max_{x \in \mathcal{X}} \|F(x)\|$.

3 Algorithms and Convergence Guarantees

In this section, we describe our algorithms for variational inequalities and state their convergence guarantees. For all of our theoretical results, we assume that the operator F is monotone. We also assume that we can perform projections onto $\mathcal{X}$. We assume that the algorithms have access to a stochastic evaluation oracle that, on input x_t, it returns a random vector $\widehat{F(x_t)}$ satisfying the following standard assumptions for a fixed (but unknown) scalar σ:

$$\mathbb{E}\left[\widehat{F(x_t)} | x_1, \ldots, x_t\right] = F(x_t) \tag{8}$$

$$\mathbb{E}\left[\left\|\widehat{F(x_t)} - F(x_t)\right\|^2\right] \leq \sigma^2 \tag{9}$$

3.1 Algorithm for Bounded Domains

Our algorithm for bounded domains is shown in Algorithm 1. Its analysis assumes that the optimization domain $\mathcal{X}$ has bounded ℓ_2-norm radius, $R = \max_{x,y \in \mathcal{X}} \|x - y\|$. The algorithm can be viewed as an adaptive version of the Past Extra-Gradient method of Popov (1980). Our update rule for the step sizes can be viewed as a generalization to the variational inequalities setting of the step sizes used by Mohri and Yang (2016); Kavis et al. (2019); Joulani et al. (2020) for convex optimization.

The following theorem states the convergence guarantees for Algorithm 1. We give the analysis in the full version. Similarly to Adagrad, setting η proportional to the radius of the domain leads to the optimal dependence on the radius and the guarantee smoothly degrades as η moves further away from the optimal choice. For simplicity, the theorem below states the convergence guarantee for $\eta = \Theta(R)$, and we give the guarantee and analysis for arbitrary η in the full version.

Theorem 3.1. *Let F be a monotone operator. Let $\eta = \Theta(R)$, where $R = \max_{x,y \in \mathcal{X}} \|x - y\|$ is the ℓ_2-diameter of the domain. Let $\overline{x}_T$ be the solution returned by Algorithm 1. If F is non-smooth, we have*

$$\mathbb{E}\left[\mathrm{Err}(\overline{x}_T)\right] \leq O\left(\frac{\gamma_0 R^2}{T} + \frac{R\left(G + \sigma\right)}{\sqrt{T}}\right)$$

where $G = \max_{x \in \mathcal{X}} \|F(x)\|$ and σ^2 is the variance parameter from assumption (9).

If F is β-smooth with respect to the ℓ_2-norm, we have

$$\mathbb{E}\left[\mathrm{Err}(\overline{x}_T)\right] \leq O\left(\frac{\left(\beta + \gamma_0\right) R^2}{T} + \frac{R\sigma}{\sqrt{T}}\right)$$

Proof. **(Sketch)** Similarly to prior works, we first upper bound the error function using the stochastic regret ($\xi_t := F(x_t) - \widehat{F(x_t)}$):

$$T \cdot \mathrm{Err}(\overline{x}_T) \leq \underbrace{\sup_{y \in \mathcal{X}} \left(\sum_{t=1}^{T} \left\langle \widehat{F(x_t)}, x_t - y \right\rangle \right)}_{\text{stochastic regret}} + \underbrace{R \left\| \sum_{t=1}^{T} \xi_t \right\| + \sum_{t=1}^{T} \left\langle \xi_t, x_t - x_0 \right\rangle}_{\text{stochastic error}}$$

Next, we analyze the stochastic regret. We split the regret into three terms and analyze each term separately:

$$\begin{aligned} &\left\langle \widehat{F(x_t)}, x_t - y \right\rangle \\ &= \left\langle \widehat{F(x_t)}, z_t - y \right\rangle + \left\langle \widehat{F(x_{t-1})}, x_t - z_t \right\rangle \\ &\quad + \left\langle \widehat{F(x_t)} - \widehat{F(x_{t-1})}, x_t - z_t \right\rangle \end{aligned}$$

The first two terms can be readily upper bounded via the optimality condition for z_t and x_t. For the third term, prior works upper bound it in terms of the iterate movement via Cauchy-Schwartz and smoothness. We crucially depart from this approach, and upper bound the term using the operator value difference $\left\|\widehat{F(x_t)} - \widehat{F(x_{t-1})}\right\|^2$, which can be significantly smaller than the iterate movement, especially in the initial iterations. Using the resulting bound on the regret, we obtain

$$T \cdot \mathrm{Err}(\overline{x}_T) \le \underbrace{O(R)\sqrt{\sum_{t=1}^{T}\left\|\widehat{F(x_t)} - \widehat{F(x_{t-1})}\right\|^2}}_{\text{loss}}$$
$$\underbrace{-\frac{1}{2}\sum_{t=1}^{T}\gamma_{t-1}\left(\|x_t - z_{t-1}\|^2 + \|x_{t-1} - z_{t-1}\|^2\right)}_{\text{gain}}$$
$$\underbrace{+ R\left\|\sum_{t=1}^{T}\xi_t\right\| + \sum_{t=1}^{T}\langle \xi_t, x_t - x_0\rangle}_{\text{stochastic error}} + \frac{1}{2}R^2\gamma_0$$

Next, we upper bound the net loss. For non-smooth operators, we ignore the gain and simply upper bound the loss by $O(G\sqrt{T})$ plus an additional stochastic error term. For smooth operators, we crucially use the gain to offset the loss. Using a careful and involved analysis, we upper bound the net loss by $O(\beta R^2)$ plus an additional stochastic error term.

Finally, we upper bound the expected stochastic error. We do so by leveraging the martingale assumption (8) and the variance assumption (9), and show that the expected error is $O(\sigma\sqrt{T})$. □

Comparison with prior work: Compared to the prior works (Bach and Levy 2019; Ene, Nguyen, and Vladu 2021), our algorithms set the step sizes based on the operator value differences instead of the iterate movement. This choice is key to obtaining optimal convergence guarantees in all settings and optimal dependencies on all of the problem parameters, matching the non-adaptive algorithms. Prior works attain convergence rates that are suboptimal by a $\Omega\left(\sqrt{\ln T}\right)$ factor (Table 1). Moreover, the prior algorithms use the off-by-one iterate (McMahan 2017), which is unavoidable due to the use of the iterate movement in the step size. These works address the off-by-one issue using additional assumptions and pay additional error terms in the convergence. Specifically, Bach and Levy (2019) require the assumption that $G := \max_{x\in\mathcal{X}} \|F(x)\|$ is bounded even when F is smooth. The algorithm requires an estimate for G in order to set the step size. Additionally, the convergence guarantee has additional error terms, including an error term of at least G^2/γ_0. In the stochastic setting, the analysis requires the stochastic operators to be bounded almost surely by a parameter $\widehat{G}$, and the algorithm requires an estimate for $\widehat{G}$ in order to set the step size. The algorithm and analysis of Ene, Nguyen, and Vladu (2021) requires knowing the radius R in order to address the off-by-one issue. The algorithm of Ene, Nguyen, and Vladu (2021) scales the update by R to ensure that the step sizes increase by at most a constant factor, and the analysis breaks if this is not ensured.

In contrast, Algorithm 1 does not suffer from the off-by-one issue. Our analysis for smooth operators does not require the operator norms to be bounded. Our convergence guarantee has optimal dependence on T and all problem parameters. Moreover, in the stochastic setting, our analysis relies only on the variance assumption (9), which is a weaker assumption than the stochastic operators being bounded almost surely.

Compared to standard methods such as the Past Extra-Gradient method (Popov 1980), our algorithms use an additional term $(\gamma_t - \gamma_{t-1})\|u - x_t\|^2$ in the update rule for z_t. Our analysis framework is versatile and allows us to analyze several variants of the algorithm, including variants that do not include this additional term. We discuss the variants and provide experimental results in the full version. The additional term leads to a tighter analysis with optimal dependencies on all problem parameters and improved constant factors. The algorithm variants performed similarly in our experiments involving bilinear saddle point problems. Our analysis readily extends to the 2-call variants of the algorithms based on the Extra-Gradient algorithm (Korpelevich 1976). In the full version, we discuss the 2-call variants and give experimental results. In all of the experiments, the 1-call algorithms performed equally well or better than their 2-call counterparts.

Our algorithm and analysis allows us to set γ_0 and η to arbitrary constants, analogous to how adaptive algorithms such as Adagrad are implemented and used in practice (γ_0 is analogous to the ϵ paramater for Adagrad). For example, the implementation of Adagrad in pytorch sets $\epsilon = 10^{-10}$ and $\eta = 0.01$. In contrast, previous works (Bach and Levy 2019; Ene, Nguyen, and Vladu 2021) need the initial value γ_0 to be at least the maximum operator norm or the radius of the domain. Moreover, the analysis of Ene, Nguyen, and Vladu (2021) does not allow the algorithm to be used with a base learning rate $\eta \neq \Theta(R)$: as noted above, the algorithm needs to scale the update by the radius to ensure that the step sizes increase by at most a constant factor, and the analysis breaks if this is not ensured.

3.2 Algorithm for Unbounded Domains

Our algorithm for unbounded domains is shown in Algorithm 2. The algorithm uses the distance from the initial point x_0 that ensures that the iterates do not diverge. The approach is inspired by the work of Fang et al. (2020) for online convex optimization, which used the distance to x_0 to stabilize mirror descent in the setting where the step sizes are chosen non-adaptively (the algorithm of Fang et al. (2020) uses the step size for the future iteration $t+1$ to perform the update for the current iteration t).

To the best of our knowledge, this is the first adaptive method for general unbounded domains, even in the special case of convex minimization. The convergence guarantees of existing adaptive algorithms in the Adagrad family depends on the maximum distance $\|x_t - x^*\|$ between the iterates and the unconstrained optimum (a point x^* with

$\nabla f(x^*) = 0$). Since these distances could diverge if the domain is unbounded, an approach employed in prior work (e.g., (Levy 2017)) is to project the iterates onto a bounded domain containing x^* (such as a ball). The resulting algorithms require access to an optimization domain containing the unconstrained optimum which may not be available or requires additional tuning (for example, for $\mathcal{X} = \mathbb{R}^d$, the distance $\|x_0 - x^*\|$ is an unknown parameter that we would need to tune in order to restrict the optimization to a ball centered at x_0 that contains x^*). Moreover, the restriction that the optimization domain contains the unconstrained optimum limits the applicability of the algorithms, as it does not allow for arbitrary constrains. Additionally, our algorithms readily extend to the more general setting of Bregman distances. Even if the domain $\mathcal{X}$ is bounded, the Bregman distances are potentially unbounded (e.g., KL-divergence distances on the simplex), and previous adaptive methods cannot be applied.

The following theorem states the convergence guarantees for Algorithm 2. We give the analysis in the full version. As before, for simplicity, the theorem states the guarantees when η is set optimally, and we give the guarantee and analysis for arbitrary η in the full version. In contrast to Algorithm 1, Algorithm 2 has the off-by-one iterate (see the discussion above) and we incur an additional error term. In the following theorem, to allow for a direct comparison with (Bach and Levy 2019), we assume that the operator norms are bounded even for smooth operators. We note that this assumption is not necessary, and we give an alternate guarantee in the full version.

Theorem 3.2. *Let F be a monotone operator. Let $D > 0$ be any fixed positive value. Let $\eta = \Theta(D)$. Let $\overline{x}_T$ be the solution returned by Algorithm 2. If F is non-smooth, we have*

$$\mathbb{E}\left[\mathrm{Err}_D(\overline{x}_T)\right] \leq O\left(\frac{\gamma_0 D^2 + \gamma_0^{-1} G^2}{T} + \frac{DG + \left(D + \gamma_0^{-1}\right)\sigma}{\sqrt{T}}\right)$$

where $G = \max_{x \in \mathcal{X}} \|F(x)\|$ and σ^2 is the variance parameter from assumption (9).

If F is β-smooth with respect to the ℓ_2-norm, we have

$$\mathbb{E}\left[\mathrm{Err}_D(\overline{x}_T)\right] \leq O\left(\frac{(\beta + \gamma_0) D^2 + DG + \gamma_0^{-1} G^2}{T} + \frac{\left(D + \gamma_0^{-1}\right)\sigma}{\sqrt{T}}\right)$$

Contemporaneous work: Antonakopoulos, Belmega, and Mertikopoulos (2021) propose to use adaptive step sizes based on operator value differences for the Extra-Gradient method, and return the weighted average of the iterates with the weights given by the step sizes. In contrast to our work, their algorithm and analysis does not extend to per-coordinate step sizes or the stochastic setting, and the convergence rate is sub-optimal by a $\Omega(\ln T)$ factor and has higher dependencies on the problem parameters (Table 1). We note that Theorem 3.2 states the convergence in terms of G, so that it can be directly compared to (Bach and Levy 2019). The stated bound is incomparable to (Antonakopoulos, Belmega, and Mertikopoulos 2021) in the smooth setting. However, our analysis can also be used to provide a bound in the same spirit in the full version).

3.3 Extensions

Our analysis framework is versatile and it allows us to analyze several variants and extensions of our main algorithms. In the full version, we consider the extension to the 2-call versions of our algorithms based on the Extra-Gradient algorithm (Korpelevich 1976). In the full version, we consider the more general setting of Bregman distances. Our analysis establishes the same convergence rate, up to constant factors. In our experimental evaluation, given in the full version, the 1-call algorithms performed equally well or better than their 2-call counterparts.

In the full version, we extend the algorithms and their analysis to the vector setting where we adaptively set a per-coordinate learning rate. The vector version of Algorithm 1 improves over the previous work of Ene, Nguyen, and Vladu (2021) by a $\Omega\left(\sqrt{\ln T}\right)$ factor (Table 1). The algorithm has optimal convergence for non-smooth operators and smooth operators that are cocoercive. For smooth operators that are not cocoercive, our convergence guarantee has a dependence of β^2 on the smoothness parameter whereas the algorithm of Ene, Nguyen, and Vladu (2021) has a better dependence of $\beta \ln \beta$. We note that, by building on the work of Ene, Nguyen, and Vladu (2021) and our approach, we can analyze a a single-call variant of the algorithm of their algorithm. For completeness, we give this analysis in the full version.

The per-coordinate methods enjoy a speed-up compared with the scalar method in many common scenarios, including learning problems with sparse gradient, as discussed in more detail in Sections 1.3 and 6 in the work of Duchi, Hazan, and Singer (2011). In our experimental evaluation, given in Section 4 and the full version, the per-coordinate methods outperformed their scalar counterparts in certain settings.

4 Experimental Evaluation

In this section, we give experimental results on bilinear saddle point instances. We provide additional experimental results, including an experiment on training generative adversarial networks, in the full version.

Instances: We consider bilinear saddle point problems $\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} f(u, v)$, where

$$f(u, v) = \frac{1}{n} \sum_{i=1}^{n} u^\top \mathbf{A}^{(i)} v$$

and $\mathbf{A}^{(i)} \in \mathbb{R}^{d \times d}$ for each $i \in [n]$. The strong solution is $x^* = (u^*, v^*) = 0$. Each matrix $\mathbf{A}^{(i)}$ was generated by first sampling a diagonal matrix with entries drawn from the $\mathrm{Uniform}([-10, 10])$ distribution, and then applying a random rotation drawn from the Haar distribution. The initial point x_0 was generated by sampling each entry from

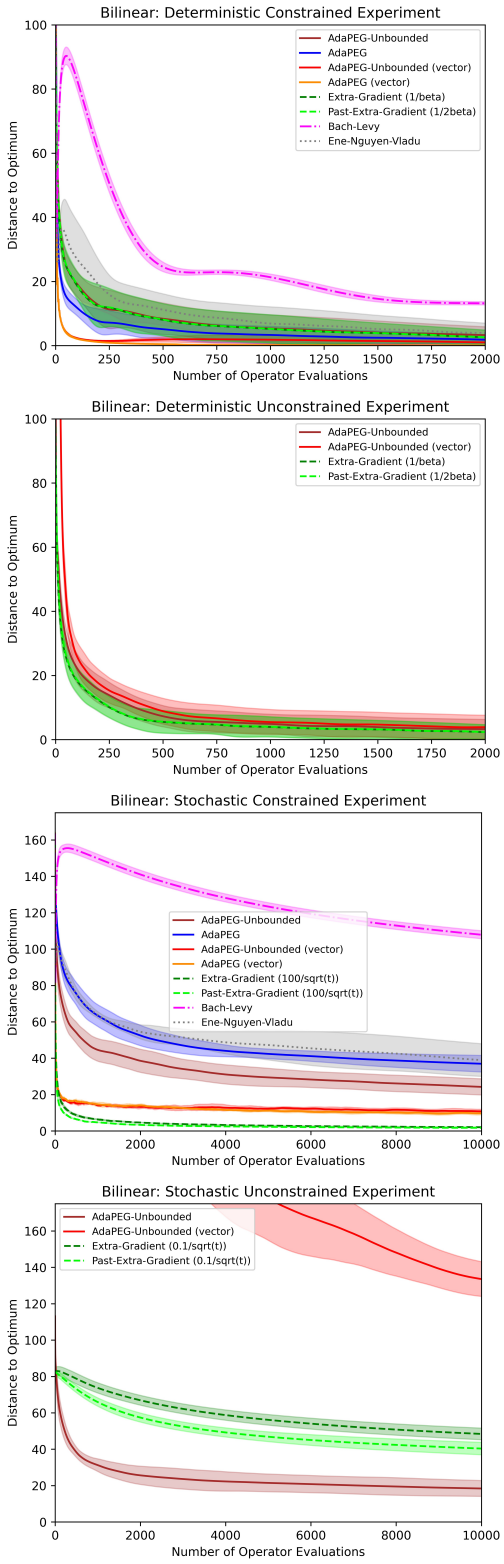

Figure 1: Convergence on bilinear instances. We report the mean and standard deviation over 5 runs.

the $\mathrm{Uniform}([-10, 10])$ distribution. We used $d = 100$ in all experiments. In the deterministic experiments, we used $n = 1$. In the stochastic experiments, we used $n = 100$ and a minibatch of size 16 for computing the stochatic evaluations. In the unconstrained experiments, the feasible domain is $\mathcal{X} = \mathcal{U} \times \mathcal{V} = \mathbb{R}^{2d}$. In the constrained experiments, $\mathcal{X} = \mathcal{U} \times \mathcal{V}$ is an ℓ_2-ball of radius $R = 2\|x_0 - x^*\|$ centered at $x^* = 0$.

Algorithms: We compare the following algorithms: our algorithms with scalar step sizes (Algorithms 1 and 2) and per-coordinate step sizes (in the full version), the adaptive methods of Bach and Levy (2019) and Ene, Nguyen, and Vladu (2021), and the non-adaptive methods Extra-Gradient (Korpelevich 1976) and Past Extra-Gradient (Popov 1980).

An experimental comparison between the 1-call algorithms and their 2-call variants can be found in the full version. In all of the experiments, the 1-call algorithms performed equally well or better than their 2-call counterparts.

We also include in the full version experimental results that include variants of our algorithms that do not include the extra term $\|u - x_t\|^2$ in the update rule for z_t. We observe that the algorithm variants perform similarly in the experiments with bounded feasible domain. We also evaluated the algorithm variants in the unconstrained setting, although this is not supported by theory. We observe that one of the variants performs slightly better in the unconstrained stochastic setting.

Hyperparameters: In the deterministic experiments, we used a uniform step size $\eta = \frac{1}{\beta}$ for the Extra-Gradient method and $\eta = \frac{1}{2\beta}$ for the Past Extra-Gradient method, as suggested by the theoretical analysis (Hsieh et al. 2019). We observed in our experiments that the additional factor of 2 is neccessary for the Past Extra-Gradient method, and the algorithm did not converge when run with step sizes larger than $\frac{1}{2\beta}$. In the stochastic experiments, we used decaying step sizes $\eta_t = \frac{c}{\sqrt{t}}$ for Extra-Gradient and Past Extra-Gradient, where c was set via a hyperparameter search. We set the parameter G_0 used by the algorithm of Bach and Levy (2019) via a hyperparameter search. For our algorithms, we set the parameter γ_0 via a hyperparameter search, and we set $\eta = R$ in the constrained experiments and $\eta = \|x_0 - x^*\|$ in the unconstrained experiments. All of the hyperparameter searches picked the best value from the set $\{1, 5\} \times \{10^5, 10^4, \ldots, 10^1, 1, 10^{-1}, \ldots, 10^{-4}, 10^{-5}\}$.

Results: The results are shown in Figure 1. We report the mean and standard deviation over 5 runs. We note that our algorithms have the best performance among the adaptive methods. Moreover, our algorithms' performance was competitive with the non-adaptive methods that have access to the smoothness parameter.

Acknowledgments

AE was supported in part by NSF CAREER grant CCF-1750333, NSF grant CCF-1718342, and NSF grant III-1908510. HN was supported in part by NSF CAREER grant CCF-1750716 and NSF grant CCF-1909314.

References

Antonakopoulos, K.; Belmega, V.; and Mertikopoulos, P. 2021. Adaptive Extra-Gradient Methods for Min-Max Optimization and Games. In *International Conference on Learning Representations (ICLR)*.

Bach, F.; and Levy, K. Y. 2019. A Universal Algorithm for Variational Inequalities Adaptive to Smoothness and Noise. In *Conference on Learning Theory (COLT)*, volume 99 of *Proceedings of Machine Learning Research*, 164–194. PMLR.

Bertsekas, D.; Nedic, A.; and Ozdaglar, A. 2003. Convex analysis and optimization, ser. *Athena Scientific optimization and computation series. Athena Scientific*.

Chambolle, A.; and Pock, T. 2011. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.*, 40(1): 120–145.

Chavdarova, T.; Gidel, G.; Fleuret, F.; and Lacoste-Julien, S. 2019. Reducing Noise in GAN Training with Variance Reduced Extragradient. In *Advances in Neural Information Processing Systems (NeurIPS)*, 391–401.

Cui, S.; and Shanbhag, U. V. 2016. On the analysis of reflected gradient and splitting methods for monotone stochastic variational inequality problems. In *IEEE Conference on Decision and Control (CDC)*, 4510–4515. IEEE.

Daskalakis, C.; Ilyas, A.; Syrgkanis, V.; and Zeng, H. 2018. Training GANs with Optimism. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Ene, A.; and Nguyen, H. L. 2021. Adaptive and Universal Algorithms for Variational Inequalities with Optimal Convergence. *CoRR*, abs/2010.07799.

Ene, A.; Nguyen, H. L.; and Vladu, A. 2021. Adaptive Gradient Methods for Constrained Convex Optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Fang, H.; Harvey, N.; Portella, V.; and Friedlander, M. 2020. Online mirror descent and dual averaging: keeping pace in the dynamic case. In *International Conference on Machine Learning (ICML)*, 3008–3017.

Gidel, G.; Berard, H.; Vignoud, G.; Vincent, P.; and Lacoste-Julien, S. 2019. A Variational Inequality Perspective on Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.

Hsieh, Y.; Iutzeler, F.; Malick, J.; and Mertikopoulos, P. 2019. On the convergence of single-call stochastic extra-gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6936–6946.

Joulani, P.; Raj, A.; Gyorgy, A.; and Szepesvari, C. 2020. A simpler approach to accelerated optimization: iterative averaging meets optimism. In *International Conference of Machine Learning (ICML)*, 4984–4993.

Juditsky, A.; Nemirovski, A.; and Tauvel, C. 2011. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1): 17–58.

Kavis, A.; Levy, K. Y.; Bach, F.; and Cevher, V. 2019. UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6257–6266.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

Korpelevich, G. 1976. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody*, 12: 747–756.

Levy, K. Y. 2017. Online to Offline Conversions, Universality and Adaptive Minibatch Sizes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1613–1622.

Malitsky, Y. V. 2015. Projected Reflected Gradient Methods for Monotone Variational Inequalities. *SIAM J. Optim.*, 25(1): 502–520.

McMahan, H. B. 2017. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18(1): 3117–3166.

McMahan, H. B.; and Streeter, M. J. 2010. Adaptive Bound Optimization for Online Convex Optimization. In *Conference on Learning Theory (COLT)*, 244–256. Omnipress.

Mertikopoulos, P.; Lecouat, B.; Zenati, H.; Foo, C.; Chandrasekhar, V.; and Piliouras, G. 2019. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.

Mohri, M.; and Yang, S. 2016. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics (AISTATS)*, 848–856.

Mokhtari, A.; Ozdaglar, A.; and Pattathil, S. 2020. Convergence Rate of $\mathcal{O}(1/k)$ for Optimistic Gradient and Extragradient Methods in Smooth Convex-Concave Saddle Point Problems. arXiv:1906.01115.

Nemirovski, A. 2004. Prox-Method with Rate of Convergence O(1/t) for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM J. Optim.*, 15(1): 229–251.

Nemirovsky, A. S. 1992. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2): 153–175.

Nemirovsky, A. S.; and Yudin, D. B. 1983. *Problem complexity and method efficiency in optimizatio*. Wiley-Interscience series in discrete mathematics. Wiley.

Nesterov, Y. 2013. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nesterov, Y. E. 2007. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Program.*, 109(2-3): 319–344.

Ouyang, Y.; and Xu, Y. 2021. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Math. Program.*, 185(1-2): 1–35.

Popov, L. D. 1980. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28(5): 845–848.

Shalev-Shwartz, S.; et al. 2011. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2): 107–194.

Yadav, A. K.; Shah, S.; Xu, Z.; Jacobs, D. W.; and Goldstein, T. 2018. Stabilizing Adversarial Nets with Prediction Methods. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.