# A hybrid algorithm for monotone variational inequalities

Rahimi Baghbadorani, R.; Mohajerin Esfahani, P.; Grammatico, S.

**Publication date**
2024

**Published in**
unknown

**Citation (APA)**
Rahimi Baghbadorani, R., Mohajerin Esfahani, P., & Grammatico, S. (2024). A hybrid algorithm for monotone variational inequalities. Manuscript submitted for publication. In *unknown*

# A hybrid algorithm for monotone variational inequalities

Reza Rahimi Baghbadorani
*Delft Center for Systems and Control*
*Delft University of Technology*
The Netherlands
R.Rahimibaghbadorani@tudelft.nl

Peyman Mohajerin Esfahani
*Delft Center for Systems and Control*
*Delft University of Technology*
The Netherlands
P.MohajerinEsfahani@tudelft.nl

Sergio Grammatico
*Delft Center for Systems and Control*
*Delft University of Technology*
The Netherlands
S.Grammatico@tudelft.nl

***Abstract*—We consider the extension of the adaptive Golden RAtio ALgorithm (aGRAAL) for variational inequalities. We show that by selecting the momentum parameter beyond the golden ratio the convergence speed can be improved, which motivates us to study the switching between small and large momentum parameters to accelerate convergence. We validate the performance of our proposed algorithms on several classes of variational inequality problems studied in the machine learning literature, including Nash equilibrium, feasibility problem, composite minimization, Markov decision processes, and zero-sum games, and compare them to existing methods.**

***Index Terms*—Variational inequality, adaptive stepsize, momentum parameter, switching algorithm.**

## I. Introduction

Variational inequality problem has recently surged into prominence in the formulation of machine learning and control problems, such as generative adversarial networks [1], robust optimization [2], [3], and optimal control [4], due to its generality. In this paper, we consider the following variational inequality (VI) problem:

$$\text{find} \quad x^* \in \mathcal{V} \quad \textbf{s.t.} \quad \langle F(x^*), x-x^*\rangle + g(x) - g(x^*) \geq 0, \quad (1)$$

for all $x \in \mathcal{V}$, where $\mathcal{V}$ is a finite-dimensional vector space. We assume that the operator $F$ is monotone and locally Lipschitz, the solution set of (1) is nonempty, and $g(x)$ is a proper convex function. Problem (1) can be written more traditionally as follows:

$$\text{find}\, x^* \in \mathcal{A} \quad \text{subject to} \quad \langle F(x^*), x - x^*\rangle \geq 0, \forall x \in \mathcal{A} \quad (2)$$

where $g(x)$ is replaced by the indicator function of the set $\mathcal{A}$ in (1). VI (1) can be considered as a general form of problems in optimization, system and control, and game theory. As an example consider the composite minimization problem $\min_{x\in\mathbb{R}^n} f(x) + g(x)$, where $f$ is a convex and smooth function and $g$ is a proper convex (and possibly nonsmooth) function. It is easy to see that by KKT condition this problem can be written as (1) with $F = \nabla f$ and the same $g(x)$ in (1) [5]. Another useful problem in optimization and control theory is min-max problem. As an example consider the following convex-concave saddle point problem $\min_{y\in\mathbb{R}^n} \max_{z\in\mathbb{R}^m} g_1(y) + f(y,z) - g_2(z)$, where $g_1(y)$ and $g_2(z)$ are the proper convex functions and $f(y,z)$ is smooth convex-concave function in respect to $y$ and $z$, respectively. By using first-order optimality condition we can rewrite this problem as (1) with the following variables

$$x = \begin{pmatrix} y \\ z \end{pmatrix}, \quad F = \begin{pmatrix} \nabla_y f(y,z) \\ -\nabla_z f(y,z) \end{pmatrix}, \quad g(x) = g_1(y) + g_2(z).$$

Last but not least in many application in reinforcement learning and game theory, we need to solve a fixed-point problem. As an example Markov decision processes (MDPs) are a powerful modeling framework in reinforcement learning, where we should solve $Tx = x$. It is not difficult to see that the equivalent form of this fixed-point problem is (1), with $F = \mathrm{Id} - T$ and $g(x) = 0$ [6].

Several iterative algorithms have been introduced to address VI problems (1). For comparison purposes, let us review some recent and closely related existing methods and for simplicity we consider (2) which is a more widely-studied problem.

**Projected Gradient descent (PGD) [7]:**

$$x^{k+1} = \pi_{\mathcal{A}}(x^k - \lambda F(x^k)).$$

where $\lambda$ is the stepsize. The convergence rate of this method is guaranteed for strongly monotone (with a strongly monotone constant $\mu$) and Lipschitz (with a Lipschitz constant $L$) operator with $\lambda \in \big(0, 2\mu/L^2\big)$.

**Extragradient descent [8]:**

$$y^k = \pi_{\mathcal{A}}(x^k - \lambda F(x^k)),$$
$$x^{k+1} = \pi_{\mathcal{A}}(x^k - \lambda F(y^k)).$$

where $\lambda$ is the stepsize, and unlike the previous method, the convergence rate is guaranteed for a Lipschitz operator (with a Lipschitz constant $L$) with $\lambda \in (0, 1/L)$. The extragradient method has been extensively studied and improved in various ways. For brevity, we refer interested readers to [8], [9] for further details.

**Projected Reflected Gradient descent (PrjRef) [10]:**

$$x^{k+1} = \pi_{\mathcal{A}}(x^k - \lambda F(2x^k - x^{k-1})).$$

where $\lambda$ is the stepsize, and the convergence rate of this method is guaranteed for a Lipschitz operator (with a Lipschitz constant

This work was supported by the ERC under the research project TRUST-91562 and COSMOS-91409.

$L$) with $\lambda \in \big(0, (\sqrt{2}-1)/L\big)$. Unlike the extragradient method, PrjRef just needs one projection per iteration.

**Golden RAtion ALgorithm (GRAAL) [11]:**

$$\begin{aligned} y^k &= (1-\beta)x^k + \beta y^{k-1}, \\ x^{k+1} &= \pi_{\mathcal{A}}(y^k - \lambda F(x^k)). \end{aligned}$$

where $\lambda$ is the step size and $\beta \in \big(0, (\sqrt{5}-1)/2\big]$. The convergence rate of this method is guaranteed for a Lipschitz operator (with a Lipschitz constant $L$) with $\lambda \in (0, 1/2\beta L)$, and it requires one projection per iteration. The step size in this method can be chosen adaptively as follows, leading to the Adaptive Golden RAtio ALgorithm (aGRAAL).

$$\lambda_k = \min\left\{(\beta+\beta^2)\lambda_{k-1}, \frac{\|x^k - x^{k-1}\|^2}{4\beta^2\lambda_{k-2}\|F(x^k) - F(x^{k-1})\|^2}, \bar{\lambda}\right\}$$

The convergence rates of all above algorithms are shown to be linear, with a difference in some constant or the time required to reach convergence.

**Contribution.** We propose two methods for solving the monotone variational inequality problem (1) that do not require the knowledge of a global Lipschitz constant. Our technical contribution is to show the ergodic $\mathcal{O}(k^{-1})$ convergence rate and R-linear rate , as in [11], under an error bound condition. In our numerical experiments, the proposed algorithm exhibits faster convergence and consistently outperforms the existing state of the art. We believe this is a testimony to the potential of switching the momentum parameter between a small and a large value, which has a significant impact on convergence speed [12]. Our results are based on the prior work by the author in [11], with the difference that we rewrite the aGRAAL algorithm using a variable momentum, within an adaptive stepsize framework. This allows us to derive conditions that support the use of a large momentum parameter, which is advantageous for convergence speed. Briefly, if $F$ is a Lipschitz and monotone operator, then our proposed method to solve (1) switches between the (likely) PGD and aGRAAL based on certain conditions, along with a stepsize rule as the aGRAAL. We note that our method rarely requires additional computations for operator evaluation and projection. However, in the worst case, we may need to perform these computations twice compared to aGRAAL.

The paper is organized as follows. In Section II, we introduce the first algorithm for solving (1) and the theoretical results are explained. Section III introduces the second adaptive method for solving (1). Finally, several illustrative examples to show the efficiency of our approach is presented in Section IV.

**Notation.** Let $\mathcal{V}$ be the finite-dimensional real vector space with the standard inner product $\langle\cdot,\cdot\rangle$ and $\ell_p$-norm $\|\cdot\|_p$ (by $\|\cdot\|$, we mean the Euclidean standard 2-norm). We also denote the $\pi_{\mathcal{A}}$ for the metric projection onto set $\mathcal{A}$ ($\pi_{\mathcal{A}}(x) = \arg\min_{y\in\mathcal{A}} \|x-y\|$), $\delta_{\mathcal{A}}$ the indicaor function of set $\mathcal{A}$, $\text{dist}(x,\mathcal{A})$ the distance from $x$ to set $\mathcal{A}$ ($\text{dist}(x,\mathcal{A}) = \|\pi_{\mathcal{A}}(x) - x\|$), and $\mathbb{B}(\tilde{x}, r)$ a closed ball with center $\tilde{x}$ and radius $r > 0$. The operator $F$ is $L$-Lipschitz, if there is $L > 0$ such that for all $x, y \in \mathcal{V}$ following inequality hold.

$$\|F(x) - F(y)\| \leq L\|x-y\| \tag{3}$$

Furthermore, $F$ is locally Lipschitz, if it is Lipschitz over any compact set of its domain. The operator $F$ is monotone if for all $x, y \in \mathcal{V}$

$$\langle F(x) - F(y), x-y\rangle \geq 0 \tag{4}$$

and it is called strongly mnotone with constant $\mu > 0$ if the following inequality holds $x, y \in \mathcal{V}$

$$\langle F(x) - F(y), x-y\rangle \geq \mu\|x-y\|^2 \tag{5}$$

We say that $F$ satisfies the Minty variational inequality problem if there exists $\hat{x} \in \mathcal{A}$ such that the following inequality holds for all $x \in \mathcal{V}$

$$\langle F(x), x-\hat{x}\rangle + g(x) - g(\hat{x}) \geq 0 \tag{Minty VI}$$

Generally (if $F$ is continuous), the solution set of the Minty VI ($\mathcal{S}^{\text{VI}}_{\text{Minty}}$) is a subset of the solution set of the main VI problem (1). The prox operator of a function $g\colon \mathcal{V} \to \mathbb{R}$ is defined as $\text{prox}_g(x) = \arg\min_u g(u) + \|u-x\|^2/2$. A function is "prox-friendly" if the prox operator is available (computationally or explicitly). The following equations and lemma are useful and commonly used in the proofs [13].

$$y = \text{prox}_g x \iff \langle y-x, z-y\rangle \geq g(y) - g(z)\, \forall z \in \mathcal{V} \tag{6a}$$

$$\begin{aligned} \|\alpha x + (1{-}\alpha)y\|^2 &= \alpha\|x\|^2 + (1-a)\|y\|^2 \\ &\quad - \alpha(1-\alpha)\|x-y\|^2 \quad \forall x, y \in \mathcal{V}, \forall \alpha \in \mathbb{R} \end{aligned} \tag{6b}$$

**Lemma I.1** (sequence convergence [11, Lemma 1]). *If $x^k \in \mathcal{V}$ is a bounded sequence, and $\lim\limits_{k\to\infty}(x^k - x)$ exists, where $x$ is a cluster point of the sequence $x^k$, then $x^k$ converges.*

## II. Preliminaries and First adaptive VI algorithm

We present our first algorithm and its convergence analysis for VI problem (1) in this section. The general form of our method is given in Algorithm 1 and 2. The algorithm follows the PGD and aGRAAL with the difference in the choice of momentum parameter, which depends on some conditions that measure the sufficient decreasing of the error bound. We first proceed with a same theorem as in [11] just with the difference that we use momentum parameter $\phi$ as a variable $\phi_k$. Before stating the theorem, let us define the merit function $\Psi(x,y) := \langle F(x), y-x\rangle + g(y) - g(x)$, which is convex with respect to $y$. It can be easily seen that (1) is equivalent to finding $x^* \in \mathcal{V}$ such that $\Psi(x^*, x) \geq 0$ for all $x \in \mathcal{V}$.

**Theorem II.1** (Analysis the adaptive golden ratio algorithm [11, Theorem 2] with variable momentum)**.** *Suppose that*

*$F\colon \operatorname{dom} g \to \mathcal{V}$ is locally Lipschitz. Then $\{x^k\}$ and $\{\bar{x}^k\}$, generated by Algorithms 1-2, satisfy the following inequality*

$$\begin{aligned}&\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^{k+1}-x\|^2+\frac{\theta_k}{2}\|x^{k+1}-x^k\|^2+2\lambda_k\Psi(x,x^k)\\&\leq\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^k-x\|^2+\frac{\theta_{k-1}}{2}\|x^k-x^{k-1}\|^2\\&-\frac{\lambda_k}{\lambda_{k-1}}\phi_k\|x^k-\bar{x}^k\|^2+\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k-1-\frac{1}{\phi_{k+1}}\big)\|x^{k+1}-\bar{x}^k\|^2\\&-\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k-\theta_k\big)\|x^{k+1}-x^k\|^2.\end{aligned}$$

*Proof.* Let $x$ be arbitrary. Now consider Algorithms 1 and 2, where $x^k$ and $\bar{x}^k$ are updated as follows

$$\bar{x}^k=\frac{(\phi_k-1)x^k+\bar{x}^{k-1}}{\phi_k},\qquad x^{k+1}=\operatorname{prox}_{\lambda_k g}(\bar{x}^k-\lambda_kF(x^k)).$$

Then by using (6a) we have

$$\langle x^{k+1}-\bar{x}^k+\lambda_kF(x^k),x-x^{k+1}\rangle\geq\lambda_k\left(g(x^{k+1})-g(x)\right)\tag{7}$$

$$\langle x^k-\bar{x}^{k-1}+\lambda_{k-1}F(x^{k-1}),x^{k+1}-x^k\rangle\geq\lambda_{k-1}\left(g(x^k)-g(x^{k+1})\right)\tag{8}$$

Multiplying (8) by $\frac{\lambda_k}{\lambda_{k-1}}\geq 0$ and using that $x^k-\bar{x}^{k-1}=\phi_k(x^k-\bar{x}^k)$, we obtain

$$\begin{aligned}&\langle\frac{\lambda_k}{\lambda_{k-1}}\phi_k(x^k-\bar{x}^k)+\lambda_kF(x^{k-1}),x^{k+1}-x^k\rangle\\&\qquad\geq\lambda_k\left(g(x^k)-g(x^{k+1})\right).\end{aligned}\tag{9}$$

Summation of (7) and (9) gives us

$$\begin{aligned}&\langle x^{k+1}-\bar{x}^k,x-x^{k+1}\rangle+\frac{\lambda_k\phi_k}{\lambda_{k-1}}\langle x^k-\bar{x}^k,x^{k+1}-x^k\rangle\\&+\lambda_k\langle F(x^k)-F(x^{k-1}),x^k-x^{k+1}\rangle\\&\geq\lambda_k\langle F(x^k),x^k-x\rangle+\lambda_k\left(g(x^k)-g(x)\right)\\&\geq\lambda_k\Big[\langle F(x),x^k-x\rangle+g(x^k)-g(x)\Big]=\lambda_k\Psi(x,x^k).\end{aligned}\tag{10}$$

Expressing the first two terms in (10) through norms leads to

$$\begin{aligned}\|x^{k+1}-x\|^2\leq{}&\|\bar{x}^k-x\|^2-\|x^{k+1}-\bar{x}^k\|^2\\&+2\lambda_k\langle F(x^k)-F(x^{k-1}),x^k-x^{k+1}\rangle\\&+\frac{\lambda_k}{\lambda_{k-1}}\phi_k\left(\|x^{k+1}-\bar{x}^k\|^2-\|x^{k+1}-x^k\|^2-\|x^k-\bar{x}^k\|^2\right)\\&-2\lambda_k\Psi(x,x^k).\end{aligned}\tag{11}$$

Similarly to (6b), we have

$$\begin{aligned}\|x^{k+1}-x\|^2={}&\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^{k+1}-x\|^2\\&-\frac{1}{\phi_{k+1}-1}\|\bar{x}^k-x\|^2+\frac{1}{\phi_{k+1}}\|x^{k+1}-\bar{x}^k\|^2.\end{aligned}\tag{12}$$

Combining this with (11), we obtain

$$\begin{aligned}&\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^{k+1}-x\|^2\leq\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^k-x\|^2+\\&\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k-1-\frac{1}{\phi_{k+1}}\big)\|x^{k+1}-\bar{x}^k\|^2-2\lambda_k\Psi(x,x^k)\\&-\frac{\lambda_k}{\lambda_{k-1}}\phi_k\big(\|x^{k+1}-x^k\|^2+\|x^k-\bar{x}^k\|^2\big)\\&+2\lambda_k\langle F(x^k)-F(x^{k-1}),x^k-x^{k+1}\rangle.\end{aligned}\tag{13}$$

Using the stepsize updating rule, the last term on the right-hand side of (13) can be upper bounded by

$$\begin{aligned}2\lambda_k\langle F(x^k)-F(x^{k-1}),x^k-x^{k+1}\rangle&\leq\\2\lambda_k\|F(x^k)-F(x^{k-1})\|\|x^k-x^{k+1}\|&\leq\\\sqrt{\theta_k\theta_{k-1}}\|x^k-x^{k-1}\|\|x^k-x^{k+1}\|&\leq\\\frac{\theta_k}{2}\|x^{k+1}-x^k\|^2+\frac{\theta_{k-1}}{2}\|x^k-x^{k-1}\|^2&.\end{aligned}\tag{14}$$

Applying the obtained estimation to (13), we deduce

$$\begin{aligned}&\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^{k+1}-x\|^2+\frac{\theta_k}{2}\|x^{k+1}-x^k\|^2+2\lambda_k\Psi(x,x^k)\leq\\&\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^k-x\|^2+\frac{\theta_{k-1}}{2}\|x^k-x^{k-1}\|^2-\frac{\lambda_k}{\lambda_{k-1}}\phi_k\|x^k-\bar{x}^k\|^2\\&+\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k-1-\frac{1}{\phi_{k+1}}\big)\|x^{k+1}-\bar{x}^k\|^2\\&-\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k-\theta_k\big)\|x^{k+1}-x^k\|^2.\end{aligned}\tag{15}$$

□

By controlling the right-hand-side of (15), we can proof the boundedness and convergence of the sequence $\{x^k\}$. Next, we aim to maintain the negativity of the last three terms of the right-hand-side of (15) while ensuring that $\phi_k$ attains a sufficiently large value which makes $\bar{x}^k$ closer to the current iterate $x^k$ instead of $\bar{x}^{k-1}$. Subsequently, we elaborate on two methods devised for achieving this objective.

**Method 1 (Algorithm 1).** In this method, we alternate between the algorithm with the small $\phi\in\left(1,\frac{1+\sqrt{5}}{2}\right]$ and the one without momentum (or equivalently $\phi=\infty$) based on residual evaluation. Recognizing that the use of the small $\phi$ ultimately leads to the convergence of the residual to zero due to the negativity of the three rightmost terms in (15) [11]. We initiate the algorithm without the momentum term, and by computing the residual ($J_k=\|x^k-\operatorname{prox}_g(x^k-F(x^k))\|$) in each iteration, we continue without $\phi$ if the residual is decreasing. Conversely, if the residual is not decreasing, we switch $\phi$ to the small value until the residual becomes smaller than the minimum residual achieved so far plus $\frac{1}{\bar{k}}$, where $\bar{k}$ denotes the iteration index

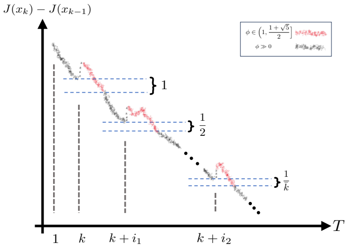


Fig. 1: The procedure of second method.

indicating the number of iterations when switching occurs. It is noteworthy that the use of a small $\phi$ may result in non-monotone changes in the residual, and we refrain from altering $\phi$ until the residual becomes smaller than the minimum residual achieved so far plus the non-summable term. Figure 1 illustrates how Algorithm 1 operates; the alteration of $\phi$ is observed when the residual decreases sufficiently, ensuring convergence due to the fact that $1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{k}+\cdots\to\infty$ as the number of iterations approaches infinity. By this method, we obtain the same inequality as in [11] (Eq. (35) in Theorem 2), which provides us with convergence to the solution of (1).

**Algorithm 1** Adaptive algorithm for VI (Method 1)

```
Require: Choose $x^0$, $x^1$, $\bar{\lambda} \gg 0$, $\lambda_0 > 0$, $\phi = (1, \frac{1+\sqrt{5}}{2}]$,
    $\theta_0 = 1$, $\rho = \frac{1}{\phi} + \frac{1}{\phi^2}$, flg $= 0$, $\bar{k} = 1$
For $k = 0, 1, 2, \dots$ do
    Find the stepsize:
        $\lambda_k = \min\left\{\rho\lambda_{k-1}, \frac{\phi\theta_{k-1}}{4\lambda_{k-1}}\frac{\|x^k - x^{k-1}\|^2}{\|F(x^k) - F(x^{k-1})\|^2}, \bar{\lambda}\right\}$
    if $(J(x^k) - J(x^{k-1}) > 0 \ \wedge$ flg $= 1) \ \vee \ \min\{J_i\}_{i=0}^{k-1} < J_k + 1/\bar{k}$ then
        $\bar{x}^k = \frac{(\phi-1)x^k + \bar{x}^{k-1}}{\phi}$, flg $= 0$
    else
        $\bar{x}^k = x^k$, flg $= 1$, $\bar{k} = \bar{k} + 1$
    end if
    Update the next iteration: $x^{k+1} = \operatorname{prox}_{\lambda_k g}(\bar{x}^k - \lambda_k F(x^k))$
    Update: $\theta_{k+1} = \frac{\phi\lambda_k}{\lambda_{k-1}}$
    Residual computation: $J_{k+1} = x^k - \operatorname{prox}_g(x^k - F(x^k))$
```

## III. Efficient switching algorithm

In this section, we analyze the convergence of our second method for solving (1). The general form of the second method is provided in Algorithm 2, which follows the adaptive golden ratio with a difference in the choice of momentum parameter compared to [11]. Differently from [11], the momentum parameter is not fixed, and in fact, in our numerical experience, it has a large value in many iterations, which supports the acceleration of the algorithms. Now, by employing (15), we use a simple analysis to control the right-hand side of (15) and aim to maintain the negativity of the right-hand side while ensuring that $\phi_k$ attains a sufficiently large value.

**Method 2 (Algorithm 2).** The algorithm is initiated with a large value for $\phi_k$, and the summation of (16) is computed after each iteration (with large value of $\phi_{k+1}$). If the resulting summation is negative, the algorithm proceeds with the initial large value of $\phi_k$. Conversely, if the summation is not negative, the algorithm is reset (by restarting, we mean that $x^{k+1}$ is generated by large $\phi$ and other parameters with indices $k$ are not considered as a new iteration and variables, lines 11-13 of Algorithm 2), and $\phi_k$ is chosen from the interval $\left(1, \frac{1+\sqrt{5}}{2}\right]$.

$$\begin{aligned}&\frac{\theta_{k-1}}{2}\|x^k - x^{k-1}\|^2 - \frac{\lambda_k}{\lambda_{k-1}}\phi_k\|x^k - \bar{x}^k\|^2\\&+\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k - 1 - \frac{1}{\phi_{k+1}}\big)\|x^{k+1} - \bar{x}^k\|^2\\&-\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k - \theta_k\big)\|x^{k+1} - x^k\|^2 - \frac{\theta_k}{2}\|x^{k+1} - x^k\|^2. \quad (16)\end{aligned}$$

After the restarting, the following equation is examined in each iteration (with large value of $\phi_{k+1}$)

$$\begin{aligned}-\frac{\lambda_k\phi_k}{\lambda_{k-1}}\|x^k - \bar{x}^k\|^2 + \big(\frac{\lambda_k\phi_k}{\lambda_{k-1}} - 1 - \frac{1}{\phi_{k+1}}\big)\|x^{k+1} - \bar{x}^k\|^2\\-\big(\frac{\lambda_k\phi_k}{\lambda_{k-1}} - \theta_k\big)\|x^{k+1} - x^k\|^2. \quad (17)\end{aligned}$$

If the computed summation is negative, the algorithm employs the large $\phi$ once more for the next iterations; conversely, if the summation is not negative, the algorithm persists with a small value of $\phi$. In this context, three scenarios are contemplated for Algorithm 2

(i) **Always negative summation:** In this scenario, by telescoping (15) the summation of (16) is always negative; therefore, $x^k \to x^*$ if $k \to \infty$ [11].

(ii) **Always positive summation:** In this scenario the summation in (17) is positive and according to the Algorithm 2 we always have $\phi \in (1, \frac{1+\sqrt{5}}{2}]$ which is the same algorithm as in [11].

(iii) **Switching between small and large $\phi$ :** If $\phi_k$ is small and by modifying $\phi_{k+1}$ to a larger value, (17) becomes negative, it is straightforward to adjust $\phi_{k+1}$ to a larger value in the subsequent step. Then, the inequality $\frac{\phi_{k+1}}{\phi_{k+1}-1} \le \frac{\phi_k}{\phi_k - 1}$ holds, and (15) in two steps is as follows

$$\begin{aligned}&\big(\frac{\phi_k}{\phi_k - 1} - \frac{\phi_{k+1}}{\phi_{k+1} - 1}\big)\|\bar{x}^k - x\|^2 + \frac{\phi_{k+1}}{\phi_{k+1} - 1}\|\bar{x}^k - x\|^2\\&+\frac{\theta_{k-1}}{2}\|x^k - x^{k-1}\|^2 + 2\lambda_{k-1}\Psi(x, x^{k-1})\\&\le \frac{\phi_k}{\phi_k - 1}\|\bar{x}^{k-1} - x\|^2 + \frac{\theta_{k-2}}{2}\|x^{k-1} - x^{k-2}\|^2\\&+\big(\frac{\lambda_{k-1}}{\lambda_{k-2}}\phi_{k-1} - 1 - \frac{1}{\phi_k}\big)\|x^k - \bar{x}^{k-1}\|^2\\&-\frac{\lambda_{k-1}}{\lambda_{k-2}}\phi_{k-1}\|x^{k-1} - \bar{x}^{k-1}\|^2\\&-\big(\frac{\lambda_{k-1}}{\lambda_{k-2}}\phi_{k-1} - \theta_{k-1}\big)\|x^k - x^{k-1}\|^2. \quad (18)\end{aligned}$$

$$\begin{aligned}&\frac{\phi_{k+1}}{\phi_{k+1} - 1}\|\bar{x}^{k+1} - x\|^2 + \frac{\theta_k}{2}\|x^{k+1} - x^k\|^2 + 2\lambda_k\Psi(x, x^k) \le\\&\frac{\phi_{k+1}}{\phi_{k+1} - 1}\|\bar{x}^k - x\|^2 + \frac{\theta_{k-1}}{2}\|x^k - x^{k-1}\|^2\\&-\frac{\lambda_k}{\lambda_{k-1}}\phi_k\|x^k - \bar{x}^k\|^2 + \big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k - 1 - \frac{1}{\phi_{k+1}}\big)\|x^{k+1} - \bar{x}^k\|^2\\&-\big(\frac{\lambda_k}{\lambda_{k-1}}\phi_k - \theta_k\big)\|x^{k+1} - x^k\|^2. \quad (19)\end{aligned}$$

Where in the first line of (18) we add and subtract $\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^k - x\|^2$. However, if $\phi_k$ is large and the summations of (16) is not negative, the algorithm should be reset with a smaller $\phi_k$. Let us assume we switch to the large $\phi$ in the $k^{\text{th}}$ iteration and after $i+1$ iterations, we change $\phi$ to a small value. In this case, the condition "$\text{sum}^1_{k+1} \leq 0$" in Algorithm 2 ensures that $\|\bar{x}^k - x\|^2 \geq \|\bar{x}^{k+i} - x\|^2$ while $\frac{\phi_k}{\phi_k-1} - \frac{\phi_{k+1}}{\phi_{k+1}-1} = -(\frac{\phi_{k+i}}{\phi_{k+i}-1} - \frac{\phi_{k+i+1}}{\phi_{k+i+1}-1})$. Therefore, (15) in two steps can be expressed as follows

$$
\begin{aligned}
&(\frac{\phi_{k+i}}{\phi_{k+i}-1} - \frac{\phi_{k+i+1}}{\phi_{k+i+1}-1})\|\bar{x}^{k+i} - x\|^2 \\
&+ \frac{\phi_{k+i+1}}{\phi_{k+i+1}-1}\|\bar{x}^{k+i} - x\|^2 + \frac{\theta_{k+i-1}}{2}\|x^k - x^{k+i-1}\|^2 \\
&+ 2\lambda_{k+i-1}\Psi(x, x^{k+i-1}) \leq \frac{\phi_{k+i}}{\phi_{k+i}-1}\|\bar{x}^{k+i-1} - x\|^2 \\
&+ \frac{\theta_{k+i-2}}{2}\|x^{k+i-1} - x^{k+i-2}\|^2 \\
&- \frac{\lambda_{k+i-1}}{\lambda_{k+i-2}}\phi_{k+i-1}\|x^{k+i-1} - \bar{x}^{k+i-1}\|^2 \\
&+ \big(\frac{\lambda_{k+i-1}}{\lambda_{k+i-2}}\phi_{k+i-1} - 1 - \frac{1}{\phi_{k+i}}\big)\|x^{k+i} - \bar{x}^{k+i-1}\|^2 \\
&- \big(\frac{\lambda_{k+i-1}}{\lambda_{k+i-2}}\phi_{k+i-1} - \theta_{k+i-1}\big)\|x^{k+i} - x^{k+i-1}\|^2. \quad (20)
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\phi_{k+i+1}}{\phi_{k+i+1}-1}\|\bar{x}^{k+i+1} - x\|^2 + \frac{\theta_{k+i}}{2}\|x^{k+i+1} - x^{k+i}\|^2 \\
&+ 2\lambda_{k+i}\Psi(x, x^{k+i}) \leq \frac{\phi_{k+i+1}}{\phi_{k+i+1}-1}\|\bar{x}^{k+i} - x\|^2 \\
&+ \frac{\theta_{k+i-1}}{2}\|x^{k+i} - x^{k+i-1}\|^2 \\
&- \frac{\lambda_{k+i}}{\lambda_{k+i-1}}\phi_{k+i}\|x^{k+i} - \bar{x}^{k+i}\|^2 \\
&+ \big(\frac{\lambda_{k+i}}{\lambda_{k+i-1}}\phi_{k+i} - 1 - \frac{1}{\phi_{k+i+1}}\big)\|x^{k+i+1} - \bar{x}^{k+i}\|^2 \\
&- \big(\frac{\lambda_{k+i}}{\lambda_{k+i-1}}\phi_{k+i} - \theta_{k+i}\big)\|x^{k+i+1} - x^{k+i}\|^2. \quad (21)
\end{aligned}
$$

Where in the first line of (20) we add and subtract $\frac{\phi_{k+i+1}}{\phi_{k+i+1}-1}\|\bar{x}^{k+i} - x\|^2$. Then by telescoping (15) (in both cases, whether switching from a small $\phi$ to a large one (18) and (19), or switching from a large $\phi$ to a small one (20) and (21)), we drive (22). More precisely, the conditions in line 7 of Algorithm 2 ensure that, by telescoping (15), we obtain similar terms on the right-hand side and left-hand side of successive lines of (15) (e.g., the leftmost terms in (18) and (20) can be removed by telescoping the inequalities, and we have similar terms on the right- and left-hand sides of two successive inequalities, like $\frac{\phi_{k+1}}{\phi_{k+1}-1}\|\bar{x}^k - x\|^2 + \frac{\theta_{k-1}}{2}\|x^k - x^{k-1}\|^2$ on the left-hand side of (18) and the right-hand side of (19)), which allows us to *point-wise* remove the similar terms and obtain the same inequality as in [11] (equation (35) in Theorem 2), which provides us with convergence to the solution of (1). In particular, by telescoping (15) for $T$ iterations, we have

$$
\begin{aligned}
&\frac{\phi_T}{\phi_T - 1}\|\bar{x}^T - x\|^2 + \frac{\theta_{T-1}}{2}\|x^T - x^{T-1}\|^2 + 2\sum_{i=1}^{T}\lambda_i\Psi(x, x^i) \\
&\leq \frac{\phi_2}{\phi_2 - 1}\|\bar{x}^1 - x\|^2 + \frac{\theta_0}{2}\|x^1 - x^0\|^2 + D. \qquad (22)
\end{aligned}
$$

Where $D$ is a non-positive number equal to the summation of the three negative rightmost terms in (15) for $T$ iterations. Note that $T$ in (22), unlike aGRAAL [11], is not exactly the number of projections or operator evaluations in Algorithm 2. In more detail, if we are in case (i) and always continue with large $\phi$, then the number of projections and operator evaluations is exactly $T$. In the worst-case scenario, we are in (ii), where the current sequence should regenerate with small $\phi$. In this situation, the number of projections and operator evaluations is $2T$. Finally, if the sequence is generated by switching between large and small $\phi$ (iii), then the number of projections and operator evaluations is between $T$ and $2T$. It is also worth noting that, in practice, the number of projections and operator evaluations is close to $T$ (please see the Numerical simulation results).

Similar to [11], we can prove the ergodic convergence rate and R-linear rate based on (22). The following lemmas indicate the convergence properties of our algorithms. To keep it short, we skip the proof and refer interested readers to [11] for further details.

**Lemma III.1** (Ergodic convergence [11])**.** *Let $X_k$ be the ergodic sequence $X_k = \sum_{i=1}^k \lambda_i x^i / \sum_{i=1}^k \lambda_i$ and $e_r(y) = \max_{x\in\mathcal{U}} \Psi(x, y) \quad \forall y \in \mathcal{V}$, where $\mathcal{U} = \operatorname{dom} g \cap \mathbb{B}(\tilde{x}, r)$ and $\tilde{x} \in \operatorname{dom} g$. Then, we obtain the $\mathcal{O}(k^{-1})$ convergence rate for the ergodic sequence $X_k$, where $M > 0$ is some constant that dominates the right-hand side of (22) for all $x \in \mathcal{U}$, in particular $\sum_{i=1}\lambda_i\Psi(x, x^i) \leq M$. More precisely we have*

$$
e_r(X_k) = \max_{x\in\mathcal{U}} \Psi(x, X_k) \leq \frac{\max\limits_{x\in\mathcal{U}} \left(\sum_{i=1}^k \Psi(x, x^i)\right)}{\sum_{i=1}^k \lambda_i} \leq \frac{M}{\sum_{i=1}^k \lambda_i}.
$$

**Lemma III.2** (R-linear convergence [11, Theorem 3])**.** *Suppose that the following error bound holds.*

$$
\textit{dist}(x, x^*) \leq \mu\|J(x,\lambda)\| \quad \forall x \in \mathcal{V}, \ \textit{with} \ \|J(x,\lambda)\| \leq \eta.
$$

*where $J(x,\lambda) = x - \operatorname{prox}_{\lambda g}(x - \lambda F(x))$ is the residual, and $\mu$ and $\gamma$ are positive constants. Then $\{x^k\}$, generated by Algorithm 1 and 2 with (23) instead of $L_1$ and $L_2$, converges to a solution of (1) at least R-linearly, where $\delta \in (0,1)$.*

$$
\lambda_k = \min\left\{\rho\lambda_{k-1}, \frac{\phi\delta\theta_{k-1}}{4\lambda_{k-1}}\frac{\|x^k - x^{k-1}\|^2}{\|F(x^k) - F(x^{k-1})\|^2}, \bar{\lambda}\right\}. \quad (23)
$$

Algorithms 1 and 2 can also be used to solve the non-monotone variational inequality problem by assuming the non-emptiness of $\mathcal{S}^{\text{VI}}_{\text{Minty}}$. The following lemma summarizes this statement.

**Lemma III.3** (Beyond monotonicity [11, Theorem 6]). *If $F$ is a locally Lipschitz and continuous operator and $g$ is a convex function, and $\mathcal{S}^{VI}_{\mathit{Minty}} \neq \varnothing$, then the sequence $x^k$ generated by Algorithms 1 and 2 converges to the solutions of* (Minty VI).

**Algorithm 2** Adaptive algorithm for VI (Method 2)

```
Require: Choose $x^0$, $x^1$, $\lambda_0 > 0$, $\bar{\lambda} \gg 0$, $\alpha = (1, \frac{1+\sqrt{5}}{2}]$,
    $\theta_0 = 1$, $\rho = \dfrac{1}{\alpha} + \dfrac{1}{\alpha^2}$, $\bar{\phi} \gg \frac{1+\sqrt{5}}{2}$, $\text{sum}^1_0 = 0$, $\text{sum}^2_0 = 0$,
    flg = 1.
 1: For $k = 0, 1, 2, \dots$ do
 2:   Find the stepsize:
        $\lambda_k = \min\left\{\rho\lambda_{k-1}, \dfrac{\alpha\theta_{k-1}}{4\lambda_{k-1}} \dfrac{\|x^k - x^{k-1}\|^2}{\|F(x^k) - F(x^{k-1})\|^2}, \bar{\lambda}\right\}$
 3:   $\bar{x}^k = \dfrac{(\phi_k - 1)x^k + \bar{x}^{k-1}}{\phi_k}$
 4:   Update the next iteration:
        $x^{k+1} = \text{prox}_{\lambda_k g}(\bar{x}^k - \lambda_k F(x^k))$
 5:   Update:   $\theta_{k+1} = \dfrac{\alpha\lambda_k}{\lambda_{k-1}}$
 6:   compute the following summations with $\phi_{k+1} = \bar{\phi}$:
        $\text{sum}^1_{k+1} = \text{sum}^1_k$ + (16)
        $\text{sum}^2_{k+1} = \text{sum}^2_k$ + (17)
 7:   if ($\text{sum}^1_{k+1} \leq 0 \ \wedge$ flg $= 1$) $\vee$ ($\text{sum}^2_{k+1} \leq 0 \ \wedge$ flg $= 0$)
      then
 8:     $\phi_{k+1} = \bar{\phi}$, flg $= 1$
 9:   else
10:     if flg $= 1$ then
11:       $x^{k+1} = x^k$, $\bar{x}^k = \bar{x}^{k-1}$
12:       $\phi_{k+1} = \alpha$, $\theta_k = \theta_{k-1}$, $\lambda_k = \lambda_{k-1}$
13:       $\text{sum}^1_{k+1} = 0$, $\text{sum}^2_{k+1} = 0$, flg $= 0$
14:     else
15:       $\phi_{k+1} = \alpha$
16:       $\text{sum}^2_{k+1} = \text{sum}^2_k$ + ((17) with $\phi_{k+1} = \alpha$)
17:       $\text{sum}^1_{k+1} = 0$
18:     end if
19:   end if
```

Algorithm 2 consists of three parts. Line 8 handles either the case of (i) or modifies $\phi_{k+1}$ to a large value (case (iii)). In lines 11-13, the algorithm adjusts $\phi_{k+1}$ to a small value (case (iii)). Finally, lines 15-17 correspond to case (ii), where the algorithm continues by updating $\text{sum}^2_{k+1}$ with a small $\phi_{k+1}$ if $\text{sum}^2_{k+1}$ is non-negative with a large $\phi_k$ and flg $= 0$.

## IV. Numerical simulations

We demonstrate the performance of Algorithm 1 and Algorithm 2 on several classes of problems studied in the literature: (1) Nash–Cournot equilibrium, (2) feasibility problem (finding an point in the intersection of balls) (3) sparse logistic regression (4) skew symmetric operator (6) Markov decision processes (5) Two-player Zero Sum Game (7) strongly monotone operator with equality constraint and (8) VI with non-monotone operator. To evaluate our performance, we compare our proposed algorithms (Algorithm 1 and Algorithm 2) with the following methods from the literature: (i) Projected Gradient descent (PrGD), (ii) projected reflected Gradient descent (PrRefGD), and (iii) adaptive Golden ratio (AdGraal), a relatively recent method for monotone variational inequality and the closest in spirit to our proposed method. Worth noting, to be more fair, we plot the residual (on the $y$-axis) against the number of operator evaluation calls (on the $x$-axis) in all our figures. In all simulation examples, we also set $\phi = 1.5$ in Algorithm 1 and aGRAAL, and in Algorithm 2, $\bar{\phi}$ and $\alpha$ are $10^6$ and 1.5, respectively.

(1) **Nash–Cournot equilibrium problem [13].** A variational inequality that corresponds to the Nash–Cournot equilibrium is find $x^* = (x^*_1, \dots, x^*_n) \in \mathbb{R}^n_+$

$$\text{subject to} \langle F(x^*), x - x^* \rangle \geq 0, \quad \forall x \in \mathbb{R}^n_+,$$

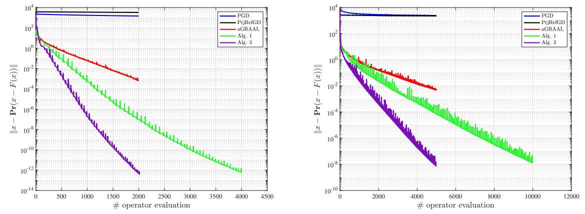
where $F(x^*) = (F_1(x^*), \dots, F_n(x^*))$ and

$$F_i(x^*) = f'_i(x^*_i) - p\Big(\sum_{j=1}^{n} x^*_j\Big) - x^*_i p'\Big(\sum_{j=1}^{n} x^*_j\Big)$$

As in [11], we assume that the function $p$ and $f_i$ are written as $p(Q) = 5000^{1/\gamma} Q^{-1/\gamma}$ and $f_i(x_i) = c_i x_i + \frac{\beta_i}{\beta_i+1} L_i^{\frac{1}{\beta_i}} x_i^{\frac{\beta_i+1}{\beta_i}}$. We set $n = 1000$ and generate our data randomly, similar to [11]. Furthermore, we consider two scenarios for each entry of $\beta$, $c$, and $L$, which are drawn independently from the uniform distributions with the following parameters

- $\gamma = 1.1$, $\beta_i \sim \mathcal{U}(0.5, 2)$, $c_i \sim \mathcal{U}(1, 100)$, $L_i \sim \mathcal{U}(0.5, 5)$;
- $\gamma = 1.5$, $\beta_i \sim \mathcal{U}(0.3, 4)$ and $c_i$, $L_i$ as above.

These parameters control the level of smoothness of $f_i$ and $p$; therefore, they can affect the convergence speed. Figure 2 reports the results where all algorithms are initialized at the same point chosen randomly. It is clear that our proposed algorithms exhibits faster convergence speed and outperforms other algorithms.



(a) $\gamma = 1.1$, $\beta_i \sim \mathcal{U}(0.5, 2)$, $c_i \sim \mathcal{U}(1, 100)$, $L_i \sim \mathcal{U}(0.5, 5)$.

(b) $\gamma = 1.5$, $\beta_i \sim \mathcal{U}(0.3, 4)$, $c_i \sim \mathcal{U}(1, 100)$, $L_i \sim \mathcal{U}(0.5, 5)$.

Fig. 2: Nash-Cournot equilibrium (1).

(2) **Feasibility problem (finding an point in the intersection of balls) [14].** In this problem we have to find a point in $x \in \cap_{i=1}^{m} C_i$, where $C_i = \mathbb{B}(c_i, r_i)$, a closed ball with a center $c_i \in \mathbb{R}^n$ and a radius $r_i > 0$. The projection onto $C_i$ is simple: $P_{C_i} x = \frac{x - c_i}{\|x - c_i\|} r_i$ if $\|x - c_i\| > r_i$

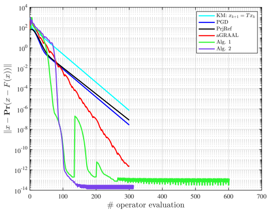


Fig. 3: Feasibility problem (intersection of balls) (2).

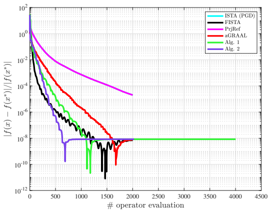


Fig. 4: Sparse logistic regression (3).

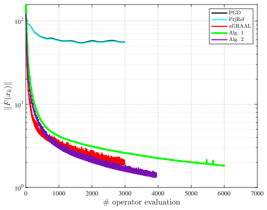


Fig. 5: Skew symmetric operator (4).

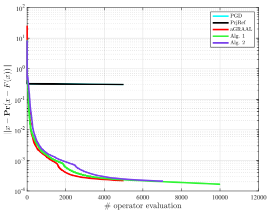


Fig. 6: Two-player Zero Sum Game (5).

and $x$ otherwise. Therefore, due to the non-expensiveness of the projection, one can use successive projections (the Krasnoselskii-Mann method) to find such a point (KM method $x^{k+1} = Tx^k = \frac{1}{m}\sum_{i=1}^m P_{C_i}x^k$) [15], [16]. On the other hand, by choosing $F = \mathrm{Id} - T$, we can rewrite this problem as a VI (1). In simulation, we choose all required parameters the same as [11]. We set $n = 1000$, $m = 2000$, the center of each ball $c_i$ is chosen randomly from $\mathcal{N}(0, 100)$, and the corresponding radius is $r_i = -\|c_i\| + 1$. The results are provided in Figure 3. The starting point of all methods are chosen as the average of all centers $c_i$.

(3) **Sparse logistic regression [17].** The sparse logistic regression can be written as follows

$$\min_x f(x) := \sum_{i=1}^m \log(1 + \exp(-b_i\langle a_i, x\rangle)) + \gamma\|x\|_1, \quad (24)$$

where $x \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$, and $b_i \in \{-1, 1\}$, $\gamma > 0$. This problem can be found in several machine learning applications, where one attempts to find a linear classifier for points $a_i$. The objective function in (24) is $f(x) = s(x) + g(x)$ with $g(x) = \gamma\|x\|_1$ and $s(x) = h(Dx)$, where matrix $D \in \mathbb{R}^{m\times n}$ as $D_{ij} = -b_i a_{ij}$ and set $h(y) = \sum_{i=1}^m \log(1 + \exp(y_i))$. It is easy to see that $s(x)$ is smooth with Lipschits constant gradient with $L_{\nabla s} = \frac{1}{4}\|D^\top D\|$. We used this constant as a stepsize in PrGD and PrRefGD. In our experiments the test data $a_i$ and $b_i$ are generated randomly using the standard Gaussian distribution, $\gamma = 0.005\|A^\top b\|_\infty$, where $A = [a_1|a_2|\cdots|a_m] \in \mathbb{R}^{n\times m}$, $n = 500$, and $m = 200$. The results are presented in Figure 4, where our methods demonstrates superior efficacy compared to other algorithms. We also plot the result of solving (24) using accelerated PrGD (FISTA) [18].

(4) **Skew symmetric operator [19, Ex. 20.35].** An example of a monotone operator that cannot satisfy even locally strong monotonicity is the skew-symmetric operator. This operator is quite simple where we have $m$ blocks of $n \times n$ skew symmetric matrices (SK), which is defined as follows

$$S = \mathrm{diag}(\mathrm{SK}_1, \mathrm{SK}_2, \cdots, \mathrm{SK}_m),\ F(x) = Sx. \quad (25)$$

In our simulation results we set $n = 10$, $m = 20$ and $\mathrm{SK}_i = \mathrm{tril}(A_i) - \mathrm{triu}(A_i)$, where $A_i$ are symmetric positive definite matrices generated randomly ($B_i = \mathrm{randn}(n, n)$, $A_i = B_i^\top B_i$). Figure 5 compares the convergence rates of different solving methods for the skew symmetric operator. As we have seen, our methods exhibit similar behavior to aGRAAL.

(5) **Two-player Zero Sum Game [20].** Generative adversarial networks (GANs) are a powerful class of neural networks that are used for unsupervised learning. The training of GANs can be considered a two-player zero sum game [1], [21]. For solving a two-player zero sum game, we need to solve the following bilinear saddle point problem,

$$\min_{x\in\Delta^m}\max_{y\in\Delta^n} \Phi(x, y) := x^\top Ay, \quad (26)$$

where $A \in \mathbb{R}^{m\times n}$ is a pay-off matrix and $\Delta^d = \{v \in \mathbb{R}^d_+ \mid \sum_{i=1}^d v_i = 1\}$ denotes the $d$-dimensional simplex. The solution of (26) is given by a saddle point $(x^*, y^*) \in \Delta^m \times \Delta^n$ satisfying

$$\Phi(x^*, y) \le \Phi(x^*, y^*) \le \Phi(x, y^*) \quad \forall (x, y) \in \Delta^m \times \Delta^n. \quad (27)$$

which can be written by problem (1) with $\mathcal{A} = \Delta^m \times \Delta^n$ and

$$F(x, y) = \begin{pmatrix} Ay \\ -A^\top x \end{pmatrix} = \begin{pmatrix} 0 & A \\ -A^\top & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \quad (28)$$

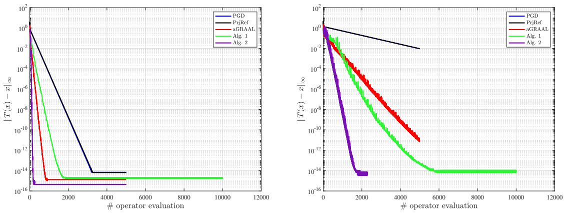
For the experiments, we set $d = m = n = 50$, and the entries of $A$ are generated with a uniform distribution on $[0, 1)$. A comparison of different methods is reported in Figure 6. As we can see from (28), this example is similar to the skew-symmetric operator (25); thus, we expect similar results.

(6) **Markov decision processes (MDPs) [22].** An MDP is a pair of $(\mathcal{S}, \mathcal{A}, \mathbb{P}, c, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state space and action space, respectively. The transition kernel $\mathbb{P}$ describes how the system moves between states: given a state $s$ and an action $a$, it shows the probability of transitioning to another state $s^+$. The cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, bounded from below, assigns a cost to each action-state pair. The discount factor $\gamma \in (0, 1)$ can be seen as a trade-off parameter between short- and long-term costs. We take $\mathcal{S} = \{1, 2, \ldots, n\}$ and $\mathcal{A} = \{1, 2, \ldots, m\}$. MDPs provide a robust modeling framework for stochastic environments, offering control mechanisms to minimize cost measures. By accessing to the transition kernel and the cost function, the problem is usually characterized by the fixed-point problem $v^\star = T(v^\star)$, i.e.,

$$v^\star(s) = [T(v^\star)](s), \quad \forall s \in \mathcal{S}, \quad (29)$$

where $T : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ is the *Bellman operator* given by

$$[T(v)](s) = \min_{a\in\mathcal{A}}\left\{c(s, a) + \gamma\mathbb{E}_{s^+\sim\mathbb{P}(\cdot|s,a)}\left[v(s^+)\right]\right\} \quad (30)$$



(a) $\gamma = 0.9$.

(b) $\gamma = 0.99$.

Fig. 7: Performance in MDP for different values of $\gamma$ (6).

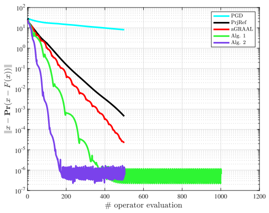


Fig. 8: Strongly monotone operator (7).

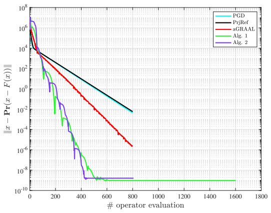


Fig. 9: Non-monotone operator (8).

That is, the optimal value function $v^\star$ is the unique fixed-point of the Bellman operator $T$. Therefore, we can reformulate this problem with (1) by $F = \mathrm{Id} - T$ and $g(x) = 0$. The comparison of the proposed algorithms in solving 50 instances of the optimal control problems of randomly generated Garnet MDPs with $n = 50$ states and $m = 5$ actions with two different values of discount factor $\gamma$ is reported in Figure 7.

(7) **Strongly monotone operator [23].** One of the popular VI problem with the strong monotone operator is (1) with linear operator $F(x) = Mx + q$, where $M$ generated randomly as $M = AA^\top + B + D$, where each entry of the $n \times n$ matrix $A$ and the skew-symmetric matrix $B$ is uniformly sampled from the interval $(-5, 5)$, and every diagonal entry of the $n \times n$ diagonal $D$ is uniformly sampled from the interval $(0, 0.3)$ (ensuring $M$ is positive definite), with each entry of $q$ uniformly sampled from $(-500, 0)$. The feasible set is $\mathcal{A} = \{x \in \mathbb{R}^n_+ \,|\, x^1 + x^2 + \cdots + x^n = n\}$. For simulation experiments, we consider $n = 100$ and choose $L = |M|$ as the Lipschitz continuity of $F$, which is used in the stepsize of PrGD and PrRefGD methods. Figure 8 illustrates the results where all algorithms are initialized at $x^0 = (1, 1, \ldots, 1)$.

(8) **Non-monotone operator [11].** As a last example, we test our proposed algorithms on a non-monotone operator mentioned in [11], where we aim to find a non-zero solution of $F(x) := M(x)x = 0$. Here, $M \colon \mathbb{R}^n \to \mathbb{R}^{n \times n}$ is a matrix-valued function, which can be considered as a VI (1) with $g = 0$. For the experiment, we define $M$ as $M(x) := t_1 t_1^\top + t_2 t_2^\top$, with $t_1 = A \sin x$, $t_2 = B \exp(x)$, where $x \in \mathbb{R}^n$, $A$ and $B \in \mathbb{R}^{n \times n}$. For the experiment, we choose $n = 500$, and the matrices $A$ and $B$ are independently and randomly generated from the normal distribution $\mathcal{N}(0, 1)$. Similar to [11], we take the initial point $x^0 = (1, 1, \ldots, 1)$. The simulation results of solving VI with the non-monotone operator $M$ are reported in Figure 9, where the proposed algorithms outperform other methods.

## References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton university press, 2009, vol. 28.

[3] M. Aghassi, D. Bertsimas, and G. Perakis, “Solving asymmetric variational inequalities via convex optimization,” *Operations Research Letters*, vol. 34, no. 5, pp. 481–490, 2006.

[4] J. C. De los Reyes, “Optimal control of a class of variational inequalities of the second kind,” *SIAM Journal on Control and Optimization*, vol. 49, no. 4, pp. 1629–1658, 2011.

[5] F. Facchinei, A. Fischer, and C. Kanzow, “Regularity properties of a semismooth reformulation of variational inequalities,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 850–869, 1998.

[6] F. E. Browder, “A new generalization of the schauder fixed point theorem,” *Mathematische Annalen*, vol. 174, no. 4, pp. 285–290, 1967.

[7] A. S. Nemirovskij and D. B. Yudin, “Problem complexity and method efficiency in optimization,” 1983.

[8] G. Korpelevich, “Extragradient method for finding saddle points and other problems,” *Matekon*, vol. 13, no. 4, pp. 35–49, 1977.

[9] P. Tseng, “On linear convergence of iterative methods for the variational inequality problem,” *Journal of Computational and Applied Mathematics*, vol. 60, no. 1-2, pp. 237–252, 1995.

[10] Y. Malitsky, “Projected reflected gradient methods for monotone variational inequalities,” *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 502–520, 2015.

[11] ——, “Golden ratio algorithms for variational inequalities,” *Mathematical Programming*, vol. 184, no. 1-2, pp. 383–410, 2020.

[12] A. Alacaoglu, A. Böhm, and Y. Malitsky, “Beyond the golden ratio for variational inequality algorithms,” *Journal of Machine Learning Research*, vol. 24, no. 172, pp. 1–33, 2023.

[13] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

[14] H. H. Bauschke and J. M. Borwein, “On projection algorithms for solving convex feasibility problems,” *SIAM review*, vol. 38, no. 3, pp. 367–426, 1996.

[15] M. A. Krasnosel’skii, “Two remarks on the method of successive approximations,” *Uspekhi matematicheskikh nauk*, vol. 10, no. 1, pp. 123–127, 1955.

[16] W. R. Mann, “Mean value methods in iteration,” *Proceedings of the American Mathematical Society*, vol. 4, no. 3, pp. 506–510, 1953.

[17] K. Mishchenko, “Regularized newton method with global convergence,” *SIAM Journal on Optimization*, vol. 33, no. 3, pp. 1440–1462, 2023.

[18] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[19] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2017.

[20] C. E. Lemke and J. T. Howson, Jr, “Equilibrium points of bimatrix games,” *Journal of the Society for industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, 1964.

[21] C. Domingo-Enrich, S. Jelassi, A. Mensch, G. Rotskoff, and J. Bruna, “A mean-field analysis of two-player zero-sum games,” *Advances in neural information processing systems*, vol. 33, pp. 20 215–20 226, 2020.

[22] M. A. S. Kolarijani and P. M. Esfahani, “From optimization to control: Quasi policy iteration,” *arXiv preprint arXiv:2311.11166*, 2023.

[23] P. T. Harker and J.-S. Pang, “For the linear complementarity problem,” *Lectures in Applied Mathematics*, vol. 26, 1990.