A Frank-Wolfe Algorithm for Strongly Monotone Variational Inequalities

Reza Rahimi Baghbadorani[1], Peyman Mohajerin Esfahani[1,2], and Sergio Grammatico[1]

Abstract. We propose an accelerated algorithm with a Frank-Wolfe method as an oracle for solving strongly monotone variational inequality problems. While standard solution approaches, such as projected gradient descent (aka value iteration), involve projecting onto the desired set at each iteration, a distinctive feature of our proposed method is the use of a linear minimization oracle in each iteration. This difference potentially reduces the projection cost, a factor that can become significant for certain sets or in high-dimensional problems. We validate the performance of the proposed algorithm on the traffic assignment problem, motivated by the fact that the projection complexity per iteration increases exponentially with respect to the number of links.

Keywords: Frank-Wolfe algorithm, projection free methods, linear minimization oracle, strongly monotone variational inequality, traffic assignment problem

1. Introduction

We consider the variational inequality problem (VIP)

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

where $\mathcal{V}$ is a compact convex set. We assume that the operator g is μ-strongly monotone, L-Lipschitz continuous, and that the solution set of (1) is nonempty. The VIP is a general framework that includes several problems and applications in systems and control theory, machine learning, and operations research. For instance, both composite convex minimization and convex-concave minimax saddle point problems can be reformulated as in (1) [3, 26], which can be used in robust controller design for systems under uncertainty [4], Nash [34] and Stackelberg games [19], supply chain optimization [1], and adversarial learning problems [14].

Fixed-point problems represent another important class in several domains, such as game theory and machine learning. Although it is straightforward to reformulate the VIP (1) as a fixed-point problem, the converse can be computationally beneficial as discussed in [26, section 3]. For instance, variational Nash equilibrium problems can be reformulated as VIPs enabling the use of efficient iterative algorithms [18, 13].

For further applications of VIPs in systems and control theory, operations research, game theory, and machine learning, we refer interested readers to [13, 36, 7, 2, 33] and references therein.

To solve the VIP (1), several iterative algorithms have been proposed. For the sake of comparison, we review some recent and closely related methods for solving the general VIP (1), where the operator g is (strongly) monotone.

Date: November 10, 2025.

The authors are with (1) Delft University of Technology and (2) the University of Toronto. This work was supported by the ERC grant TRUST-949796 and the NSERC Discovery grant RGPIN-2025-06544.

(i) **Projected Gradient Descent** [30]**:** A classical approach inspired by gradient descent in the optimization literature is

$$x_{k+1} = \mathrm{proj}_{\mathcal{V}}\Big(x_k - \alpha g(x_k)\Big),$$

where the operator $g(\cdot)$ replaces the gradient operator and α is the stepsize. In the literature of fixed point computation (e.g., dynamic programming), this is also known as *Value Iteration*. This method guarantees convergence for strongly monotone and Lipschitz operator g for any stepsize $\alpha \in (0, 2\mu/L^2)$ where μ and L are the strong monotonicity constant and the Lipschitz constant, respectively.

(ii) **Extragradient Descent** [27]**:** An improvement to the gradient descent approach is to call the operator g twice in order to improve the convergence rate. This yields the algorithm

$$\begin{aligned} y_k &= \mathrm{proj}_{\mathcal{V}}\Big(x_k - \alpha g(x_k)\Big), \\ x_{k+1} &= \mathrm{proj}_{\mathcal{V}}\Big(x_k - \alpha g(y_k)\Big), \end{aligned}$$

with α as the stepsize. Unlike classic projected gradient descent, this method does not require strong monotonicity of the operator g and ensures convergence for a Lipschitz operator when the stepsize $\alpha \in (0, 1/L)$.

(iii) **Accelerated gradient descent** [32]**:** An influential idea in optimization, first proposed by Nesterov [31], is to accelerate algorithm convergence by incorporating a so-called *momentum* into the update dynamics. One can draw a parallel, in a similar fashion as in the Gradient Descent (i.e., replacing the gradient with the operator g), and arrive at

$$\begin{aligned} x_k &= \arg\max_{x \in \mathcal{V}} \sum_{i=0}^{k} \alpha_i \left[\langle g(y_i), y_i - x\rangle - \frac{\mu}{2}\|x - y_i\|^2\right], \\ y_{k+1} &= \arg\max_{y \in \mathcal{V}} \langle g(x_k), x_k - y\rangle - \frac{\beta}{2}\|y - x_k\|^2, \end{aligned}$$

where to ensure the convergence, it suffices to choose the stepszie and momentum coefficients as $\alpha_{k+1} = \frac{\mu}{L}\sum_{i=0}^{k} \alpha_i$ and $\beta = L$.

(iv) **Projected Reflected Gradient Descent** [25]**:** Evaluating the operator at a reflected point, an extrapolation of the current and previous iterates, enhances stability and convergence in monotone VIP, leading to the following algorithm

$$x_{k+1} = \mathrm{proj}_{\mathcal{V}}\Big(x_k - \alpha g(2x_k - x_{k-1})\Big),$$

where $\alpha \in \Big(0, (\sqrt{2} - 1)/L\Big)$ is the stepsize. This method guarantees convergence for a Lipschitz operator, and unlike the extragradient method, it requires only one projection per iteration.

(v) **Golden Ratio Algorithm** [26]**:** To guarantee convergence when using the Gradient Descent method for Lipschitz but non-strongly monotone operators, negative momentum parameters are required. Introducing a negative momentum ensures convergence, leading to the algorithm

$$\begin{aligned} y_k &= (1 - \zeta)x_k + \zeta y_{k-1}, \\ x_{k+1} &= \mathrm{proj}_{\mathcal{V}}\Big(y_k - \alpha g(x_k)\Big), \end{aligned}$$

with $\alpha \in \Big(0, 1/(2\zeta L)\Big)$ as the stepsize and momentum parameter $\zeta \in \Big(0, (\sqrt{5}-1)/2\Big]$. Additionally, overcomes the limitation of methods that rely on problem constants (like Lipschitz continuity parameter); the stepsize can be chosen adaptively, leading to the adaptive golden ratio algorithm [26]

$$\alpha_k = \min\left\{(\zeta+\zeta^2)\alpha_{k-1}, \frac{\|x_k - x_{k-1}\|^2}{4\zeta^2\alpha_{k-2}\|g(x_k) - g(x_{k-1})\|^2}\right\}.$$

(vi) **Operator splitting methods** [13]**:** The operator g can be split into a summation of different operators. Solving VIP (1) in these cases is equivalent to solving the fixed-point problem $0 \in g_1(x) + g_2(x)$, where $g(x) + \mathcal{N}_{\mathcal{V}}(x) = g_1(x) + g_2(x)$ and $\mathcal{N}_{\mathcal{V}}(x)$ is the normal cone of the compact, convex set $\mathcal{V}$ at the point x. The iterative update of each sub-operator leads to convergence towards the solution. A well-known class of these algorithms is the Douglas-Rachford splitting method [5], which can be written as

$$\begin{aligned} y_{k+1} &= (I + \alpha g_2)^{-1}\Big(x_k - \alpha g_1(x_k)\Big), \\ x_{k+1} &= x_k + \zeta(y_{k+1} - x_k), \end{aligned}$$

where the stepsize can be chosen $\alpha \in (0, 1]$ and $\zeta \in (0, 2)$ is a relaxation parameter. The convergence of this method is guaranteed for different cases of g_1 and g_2; we refer interested readers to [16, 29] for further details. Additionally, we refer them to [28] for further algorithms related to applications of monotone variational inequalities and an open source Python toolbox.

In the methods mentioned above, the projection operator onto the feasible set $\mathcal{V}$ ($\text{proj}_{\mathcal{V}}(\cdot)$) is a necessary operation of the algorithm which can be costly in some cases. Frank-Wolfe (FW) is a classical approach to avoid this projection complexity by resorting to a linear oracle minimization (as opposed to the quadratic optimization of the projection operator) over the same set. Table 1 summarizes the computational complexity of the linear oracle minimization and the projection onto certain sets [11, 9].

Table 1 – Complexity of linear minimization and projection. The parameter ε denotes the precision of linear minimization or projection operator.

Set	Linear minimization	Projection
(1) n-dimensional ℓ_p-ball, $p \neq 1, 2, \infty$.	$\mathcal{O}(n)$	$\tilde{\mathcal{O}}(n/\varepsilon^2)$
(2) Nuclear norm ball of $n \times m$ matrices. θ and τ denote the number of non-zero entries and the top singular value of the projected matrix, respectively.	$\mathcal{O}(\theta \ln(m+n)\sqrt{\tau/\varepsilon})$	$\mathcal{O}(mn \min\{m, n\})$
(3) Flow polytope on a graph with m vertices and n edges with capacity bound on edges.	$\mathcal{O}((n \log m)(n + m \log m))$	$\mathcal{O}(n^4 \log n)$
(4) Birkhoff polytope ($n \times n$ doubly stochastic matrices).	$\mathcal{O}(n^3)$	$\tilde{\mathcal{O}}(n^2/\varepsilon^2)$

For concave-convex minimax problems and inspired by the recent works [20, 22, 17], the authors of [15] and [10] prove convergence of different types of FW algorithms. They demonstrate the numerical efficiency of the FW algorithm via illustrative saddle point problems, in which projecting onto the corresponding sets is computationally demanding or even intractable; see the perfect matching problem as an example of this class of problems [21]. Motivated by this, a natural question is whether the FW algorithm can also be utilized for solving the VIP (1). To the best of our knowledge, this question is still largely unexplored, which is the focus of this study.

Contribution. Following the footsteps of the accelerated Nesterov's technique for solving strongly monotone variational inequalities [32] and the FW algorithm for solving saddle point problems [15], we propose a novel accelerated algorithm with FW method as an oracle for solving the strongly monotone VIP (1) and provide a non-asymptotic convergence rate. To validate the theoretical results, we implement the proposed method in the traffic assignment problem, an important application in transportation and operation research [6, 8, 7], where the complexity of the problem (the corresponding set) increases exponentially with the number of variables.

Roadmap. The paper is organized as follows: Section 2 reviews the Frank-Wolfe technique and states lemmas for solving *minmax* saddle point problems. In Section 3, we propose our algorithm and provide the technical proofs of the convergence theorems. Section 4 benchmarks the proposed algorithm in a traffic assignment application. Finally, the conclusion and future research directions are given in Section 5.

Notation. Let $\mathcal{V}$ be a finite-dimensional real Hilbert space equipped with the standard inner product $\langle \cdot, \cdot \rangle$ and the associated norm $\|\cdot\|$ (so that we may simply write $\|x\|^2 = \langle x, x\rangle$). We define the normal cone of the set $\mathcal{V}$ at the point $x \in \mathcal{V}$ as $\mathcal{N}_{\mathcal{V}}(x) = \{u : u^\top x \geq u^\top y,\ \forall y \in \mathcal{V}\}$. The operator $\text{proj}_{x\in\mathcal{V}}$ denotes the projection onto set $\mathcal{V}$ with respect to the underlying inner product norm (i.e., $\text{proj}_{x\in\mathcal{V}}(x) := \arg\min_{y\in\mathcal{V}}\|x-y\|$). We define the diameter and boundary distance of a set $\mathcal{X}$ as

$$D_{\mathcal{X}} := \sup_{x,x'\in\mathcal{X}} \|x - x'\| \quad \text{and} \quad \sigma_{\mathcal{X}}(x) := \min_{s\in\partial\mathcal{X}} \|x - s\|.$$

The function $F(x,y)$ is $(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ strongly convex-concave if $F(x,y) - \frac{\mu_{\mathcal{X}}}{2}\|x\|^2 + \frac{\mu_{\mathcal{Y}}}{2}\|y\|^2$ is convex-concave. The constants L_{XY} and L_{YX} are the cross smooth constants of $F(x,y)$ (or equivalently the Lipschitz constants of ∇F) over the set $\mathcal{X}\times\mathcal{Y}$ if for all $x, \bar{x} \in \mathcal{X}$ and $y, \bar{y} \in \mathcal{Y}$

$$\|\nabla_x F(x,y) - \nabla_x F(x,\bar{y})\| \leq L_{XY}\|y - \bar{y}\|,$$
$$\|\nabla_y F(x,y) - \nabla_y F(\bar{x},y)\| \leq L_{YX}\|x - \bar{x}\|.$$

For brevity, we refer to $F(x,y)$ as a smooth function with constant $L_0 = \max(L_{XY}, L_{YX})$ if it is cross-smooth over $\mathcal{X}\times\mathcal{Y}$.

2. Assumptions and Technical preliminaries

Before proceeding with the solution to the VIP (1), we begin with some assumptions and lemmas that will be used throughout the paper. We note that most of the results build on the seminal work by Nesterov [32] and Jaggi [20]. The following assumptions hold throughout this study.

Assumption 2.1 (Operator regularity)**.** *We assume that the solution set of VIP* (1) *is nonempty, where* $\mathcal{V}$ *is a compact, convex set and the operator* g *satisfies the following:*

(i) L-Lipschitzness: $\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{V},$

(ii) μ-strong monotonicity: $\langle g(x) - g(y), x - y\rangle \geq \mu\|x - y\|^2, \quad \forall x, y \in \mathcal{V}.$

Next, we start with a minimax problem, a core component of our convergence analysis for VIP (1). Specifically, by leveraging the following minimax oracle at each iteration, we aim to eliminate the need for costly projection steps:

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} F(x,y). \tag{2}$$

An important concept relevant to the convergence of the *minimax* problem (2), which helps us measure the convergence rate of (2), is the error function, defined as

$$h_k := F(x_k,\hat{y}_k) - F(\hat{x}_k,y_k), \tag{3}$$

where $\hat{x}_k = \arg\max_{x\in\mathcal{X}} F(x,y_k)$ and $\hat{y}_k = \arg\max_{y\in\mathcal{Y}} F(x_k,y)$. The following proposition summarizes the convergence of (2) in terms of the error function h_k, using the FW algorithm.

Algorithm 1 FW-minimax oracle for (2) [15]

```
Let $z_0 = (x_0, y_0) \in \mathcal{X}\times\mathcal{Y}$
for $k = 0 \dots T$ do
    Compute the partial gradients of $F$: $r_k = \begin{pmatrix} \nabla_x F(x_k,y_k) \\ -\nabla_y F(x_k,y_k) \end{pmatrix}$,
    Compute the desired direction: $s_k = \arg\min_{z\in\mathcal{X}\times\mathcal{Y}} \langle z, r_k\rangle$,
    Compute the stepsize: $\alpha_k = \min\left(1, \frac{\nu}{2C}\langle z_k - s_k, r_k\rangle\right)$ or $\alpha_k = \frac{2}{2+k}$   (based on Proposition 2.2)
    Compute the next iteration: $z_k = (1-\alpha_k)z_k + \alpha_k s_k$.
end for
```

Proposition 2.2 (FW-minimax oracle convergence [15, Theorem 1])**.** *Let $F(x,y)$ be L_0-smooth continuous and $(\mu_{\mathcal{X}},\mu_{\mathcal{Y}})$ strongly convex-concave on a convex, compact set $\mathcal{X}\times\mathcal{Y}$ and (x_*,y_*), the solution of (2), is in the interior of $\mathcal{X}\times\mathcal{Y}$. Consider applying the Frank-Wolfe algorithm (Algorithms 1) for saddle point problem (2) with the stepsize*

$$\alpha_k = \min\left(1, \frac{\nu}{2C}\langle z_k - s_k, r_k\rangle\right), \tag{4}$$

where the constants are

$$\rho = \nu^2\frac{\sigma_\mu^2}{2C}, \quad C = \frac{L_0 D_{\mathcal{X}}^2 + L_0 D_{\mathcal{Y}}^2}{2},$$
$$\nu = 1 - \frac{\sqrt{2}}{\sigma_\mu}\max\left\{\frac{D_{\mathcal{X}} L_{XY}}{\sqrt{\mu_{\mathcal{Y}}}}, \frac{D_{\mathcal{Y}} L_{YX}}{\sqrt{\mu_{\mathcal{X}}}}\right\},$$
$$\sigma_\mu = \sqrt{\min(\mu_{\mathcal{X}}\sigma_{\mathcal{X}}^2(x_*), \mu_{\mathcal{Y}}\sigma_{\mathcal{Y}}^2(y_*))}.$$

Then, the convergence rate for the error h_k (3) is $\mathcal{O}\left((1-\rho)^{\frac{k}{6}}\right)$.
Moreover, if $\sigma_\mu > 2\max\left\{\frac{D_{\mathcal{X}} L_{XY}}{\mu_{\mathcal{Y}}}, \frac{D_{\mathcal{Y}} L_{YX}}{\mu_{\mathcal{X}}}\right\}$ and the stepsize is set to $\alpha_k = \frac{2}{2+k}$, then the error h_k (3) decreases at a sublinear rate of $\mathcal{O}(1/k)$.

Remark 2.3 (FW-variants)**.** *Other FW variants, including the away-step and pairwise algorithms, achieve similar convergence guarantees under alternative assumptions to those in Proposition 2.2, such as when the feasible set is a polytope. These variants often yield better practical performance by allowing corrections to previously chosen directions, which helps avoid flat regions and reduces* *zig-zagging behavior near the solution. More recently, the authors in* [10] *proposed a projection-free method under assumptions similar to those in Proposition 2.2, but without requiring the solution to lie in the interior of the feasible set and their method is based on a three-loop algorithm. The proposed FW Algorithm 1 can be replaced by these alternatives. For further details, we refer interested readers to* [15, 10].

Proposition 2.2 lays the foundation for proving the convergence of the projection-free algorithm for solving VIP (1). It avoids expensive projection steps by instead solving a strongly convex-concave *minimax* problem at each iteration, which is the central motivation behind this study.

We move forward by introducing a gap function to measure the optimality gap of VIP (1) and by proposing two lemmas that are central to the development of the algorithm in this paper.
It is not difficult to see that, in light of the strong monotonicity of the operator g, the solution of VIP (1), x^*, satisfies the following inequality

$$\langle g(y), x^* - y\rangle + \frac{\mu}{2}\|y - x^*\|^2 \leq \langle g(x^*), x^* - y\rangle - \frac{\mu}{2}\|y - x^*\|^2 \leq 0, \quad \forall\, y \in \mathcal{V}. \tag{5}$$

In order to measure the the approximated solution of (1), we introduce a gap function $f(x)$ and the following lemma.

$$f(x) := \sup_{y\in\mathcal{V}} \Big\{\langle g(y), x - y\rangle + \frac{\mu}{2}\|y - x\|^2\Big\}. \tag{6}$$

Lemma 2.4 (Gap function properties, [32, Theorem 1])**.** *The gap function $f(x)$ is a nonnegative, well-defined, μ-strongly convex function on $\mathcal{V}$ and vanishes at the unique solution of* (1)*.*

Using Lemma 2.4 and the definition (6), our goal is to minimize $f(x)$, which is equivalent to solving the VIP (1). To this end, let us define the following quantities

$$S_N := \sum_{i=0}^{N} \lambda_i, \quad \tilde{y}_N := \frac{1}{S_N}\sum_{i=0}^{N} \lambda_i y_i, \quad \Delta_N := \max_{x\in\mathcal{V}} \Bigg\{ \sum_{i=0}^{N} \lambda_i \Big[g(y_i), y_i - x\rangle - \frac{\mu}{2}\|x - y_i\|^2 \Big] \Bigg\}, \tag{7}$$

where $\{y_i\}_{i=0}^{N} \subset \mathcal{V}$ and $\{\lambda_i\}_{i=0}^{N}$ are sequences of arbitrary points and positive weights, respectively. Next lemma shows the upper bound for the gap function $f(x)$.

Lemma 2.5 (Gap function upper bound, [32, Lemma 1])**.** *Using the quantities defined in* (7)*, we have the inequality*

$$f(\tilde{y}_N) \leq \frac{1}{S_N}\Delta_N. \tag{8}$$

The proof follows from the definition of the gap function (6) and the strong monotonicity of the operator $g(x)$. For brevity, we skip the proof and refer interested readers to [32] for further details.

3. Proposed Frank-Wolfe Algorithm and Convergence Analysis

Building on the key proposition and lemmas from the previous section, this section pertains to the analysis of convergence for using the FW algorithm as an oracle in solving VIP (1). The general form of our proposed method is provided in Algorithm 2, which follows the accelerated gradient descent method [32], with the difference that the projection steps are replaced by a strongly convex-concave minimax problem solved by the FW algorithm.

Lemmas 2.4 and 2.5 shed light on the behavior of the gap function (6) and highlight the goal of minimizing and controlling the growth of Δ_N. For $\beta > 0$, consider the functions

$$\psi_y^\beta(x) := \langle g(y), y - x\rangle - \frac{\beta}{2}\|x-y\|^2, \quad \Psi_k(x) := \sum_{i=0}^{k} \lambda_i \psi_{y_i}^{\mu}(x).$$

We note that $\Delta_k = \max_{x\in\mathcal{V}} \Psi_k(x)$, and the functions $\psi_y^\beta(x)$ and $\Psi_k(x)$ are strongly concave with constants β and μS_k, respectively. Consider the following iterations

$$\begin{cases} x_k = \arg\min_{x_k\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{W(x,x_k) - \frac{\mu}{4}S_k\|x-x_k\|^2\Big\}, \\ y_{k+1} = \arg\min_{y_{k+1}\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{\langle -g(x_k) - \beta(y_{k+1}-x_k), x - y_{k+1}\rangle - \frac{\mu}{4}\|x-y_{k+1}\|^2\Big\}, \end{cases} \tag{9}$$

where

$$\begin{aligned} W(x,x_k) &:= \langle \nabla\Psi_k(x_k), x - x_k\rangle = \Big\langle \sum_{i=0}^{k}\lambda_i(-g(y_i) - \mu(x_k - y_i)), x - x_k\Big\rangle \\ &= -\mu S_k\langle x_k, x - x_k\rangle + \Big\langle \sum_{i=0}^{k}\lambda_i(-g(y_i) + \mu y_i), x - x_k\Big\rangle. \end{aligned}$$

We are now in a position to derive an upper bound for Δ_k using iterations (9), which helps us establish the convergence of the iterates to the solution of the VIP (1).

Proposition 3.1 (Upper bound of Δ_k)**.** *If $\lambda_{k+1} \le \frac{\mu}{2\beta}S_k$, then by using the iterates (9), we have*

$$\Delta_{k+1} \le \Delta_k + \lambda_{k+1}\Big[\frac{1}{\mu+2\beta}\|g(y_{k+1}) - g(x_k)\|^2 - \frac{\beta}{2}\|y_{k+1}-x_k\|^2\Big]. \tag{10}$$

Proof. We know that $\Psi_{k+1}(x) = \Psi_k(x) + \lambda_{k+1}\psi_{y_{k+1}}^{\mu}(x)$. Then, we have

$$\begin{aligned} \Delta_{k+1} &= \max_{x\in\mathcal{V}}\Big\{\Psi_k(x) + \lambda_{k+1}\psi_{y_{k+1}}^{\mu}(x)\Big\} \\ &\le \Delta_k + \max_{x\in\mathcal{V}}\Big\{\langle\nabla\Psi_k(x_k), x - x_k\rangle - \frac{\mu}{2}S_k\|x-x_k\|^2 + \lambda_{k+1}\psi_{y_{k+1}}^{\mu}(x)\Big\} \\ &\le \Delta_k + \max_{x\in\mathcal{V}}\Big\{W(x,x_k) - \frac{\mu}{4}S_k\|x-x_k\|^2\Big\} + \max_{x\in\mathcal{V}}\Big\{-\frac{\mu}{4}S_k\|x-x_k\|^2 + \lambda_{k+1}\psi_{y_{k+1}}^{\mu}(x)\Big\}. \end{aligned}$$

Note that, if we consider $x^\dagger = \arg\max_{x\in\mathcal{V}}\Psi_k(x) = \mathrm{proj}_{x\in\mathcal{V}}\left(\frac{\sum_{i=0}^{k}\lambda_i(\mu y_i - g(y_i))}{\mu S_k}\right)$, then, from the definition of the projection operator and the first-order optimality condition, we have $W(x,x^\dagger) = \langle\nabla\Psi_k(x^\dagger), x - x^\dagger\rangle \le 0$. Therefore, we can conclude that, as indicated in Proposition 2.2, the iterates x_k in (9) ensures $\min_{x_k\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{W(x,x_k) - \frac{\mu}{4}S_k\|x-x_k\|^2\Big\} < 0$. We further note that this strongly convex-concave minimax problem can be solved by FW algorithm with (sub)-linear convergence guarantee. Therefore, by the definition of x_k in (9), we arrive at

$$\begin{aligned} \Delta_{k+1} &\le \Delta_k + \overbrace{\min_{x_k\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{W(x,x_k) - \frac{\mu}{4}S_k\|x-x_k\|^2\Big\}}^{\le 0} + \max_{x\in\mathcal{V}}\Big\{-\frac{\mu}{4}S_k\|x-x_k\|^2 + \lambda_{k+1}\psi_{y_{k+1}}^{\mu}(x)\Big\} \\ &\le \Delta_k + \max_{x\in\mathcal{V}}\Big\{-\frac{\mu}{4}S_k\|x-x_k\|^2 + \lambda_{k+1}\Big[\langle g(y_{k+1}), y_{k+1} - x\rangle - \frac{\mu}{2}\|x-y_{k+1}\|^2\Big]\Big\}. \end{aligned} \tag{11}$$

Now, let us analyze the second term in the "max" part of the right-hand-side of (11). Note that

$$\langle g(y_{k+1}), y_{k+1}-x\rangle - \frac{\mu}{2}\|x-y_{k+1}\|^2 = \langle g(y_{k+1}) - g(x_k), y_{k+1}-x\rangle - \frac{\mu}{2}\|x-y_{k+1}\|^2 + \langle g(x_k), y_{k+1}-x\rangle,$$

Similar to the previous part, if $y^\dagger$ considered as $y^\dagger = \mathrm{proj}_{x\in\mathcal{V}}\left(x_k - \frac{g(x_k)}{\beta}\right)$, we have

$$\langle -g(x_k) - \beta(y^\dagger - x_k), x - y^\dagger\rangle \le 0, \quad \forall x \in \mathcal{V}.$$

Therefore, y_{k+1} in (9) ensures $\min_{y_{k+1}\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{\langle -g(x_k) - \beta(y_{k+1}-x_k), x-y_{k+1}\rangle - \frac{\mu}{4}\|x-y_{k+1}\|^2\Big\} < 0$, and using the definition of y_{k+1} we can write

$$\begin{aligned}&\langle g(y_{k+1}), y_{k+1}-x\rangle - \frac{\mu}{2}\|x-y_{k+1}\|^2\\ &\qquad\le \|g(y_{k+1}) - g(x_k)\|\|y_{k+1}-x\| - \frac{\mu}{4}\|y_{k+1}-x\|^2 + \beta\langle y_{k+1}-x_k, x-y_{k+1}\rangle,\end{aligned}$$

using above inequality and maximizing the right-hand-side based on $\|y_k - x\|$, one obtains

$$\langle g(y_{k+1}), y_{k+1}-x\rangle - \frac{\mu}{2}\|x-y_{k+1}\|^2 \le \frac{1}{\mu+2\beta}\|g(y_{k+1}) - g(x_k)\|^2 + \frac{\beta}{2}\|x-x_k\|^2 - \frac{\beta}{2}\|y_{k+1}-x_k\|^2.$$

Putting the above inequality together with (11), and using the upper bound on λ_{k+1}, we obtain the inequality (10). □

Now by setting $\lambda_{k+1} = \frac{\mu}{2L}S_k = \frac{1}{2\gamma}S_k$ in (10), where L and $\gamma = \frac{L}{\mu}$ are the Lipschitz constant and the condition number of the operator g, respectively, we have $\Delta_{k+1} < \Delta_k$. We are now ready to show the convergence rate of Algorithm 2 for solving VIP (1).

Algorithm 2 Frank-Wolfe type algorithm for the VIP (FW-VIP)

```
Given $\gamma = \frac{L}{\mu}, \lambda_0 = 1, \varepsilon$.
for $k = 0, \ldots, T$ do
    $\left\{\begin{aligned} x_k &= \arg\min_{x_k\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{W(x,x_k) - \frac{\mu}{4}S_k\|x-x_k\|^2\Big\} \\ y_{k+1} &= \arg\min_{y_{k+1}\in\mathcal{V}}\max_{x\in\mathcal{V}}\Big\{\langle -g(x_k) - L(y_{k+1}-x_k), x-y_{k+1}\rangle - \frac{\mu}{4}\|x-y_{k+1}\|^2\Big\}\end{aligned}\right\}$   FW-minimax oracle 1 with $\varepsilon$-precision
    $\lambda_{k+1} = \frac{1}{2\gamma}S_k$,
end for
output: $\tilde{y}_k = \frac{1}{S_k}\sum_{i=0}^{k}\lambda_i y_i$.
```

Theorem 3.2 (VIP-convergence via FW-minimax oracle)**.** *Consider the VIP* (1) *under Assumptions 2.1, and the FW-minimax oracle 1. Let $\gamma = L/\mu$ be the condition number of the operator g, and f the gap function defined in* (6)*. Algorithm 2 returns an ε accurate solution to* (1) *if the number of iterations is $T \ge (\gamma+1)\log\left(\gamma^2 f(x_0)/\varepsilon\right)$.*

Before proceeding with the proof of the theorem, we summarize the overall convergence complexity of using Algorithm 2 with the FW-minimax oracle 1 as follows.

Remark 3.3 (Overall complexity of VIP solution)**.** *The overall complexity of the solution to the VI problem* (1) *is the product of the complexity of the FW-minimax oracle in Algorithm 1 with the complexity of Algorithm 2, under the assumption that the solution of each sub-minimax problem lies inside the feasible domain $\mathcal{V}$. In particular, by using the stepsize* (4) *in Algorithm 1, resulting in*

$$\underbrace{\mathcal{O}\left(\log\left(1/\varepsilon\right)\right)}_{\textit{Theorem 3.2}} \cdot \underbrace{\mathcal{O}\left(\log\left(1/\varepsilon\right)\right)}_{\textit{Proposition 2.2}} = \mathcal{O}\left(\log^2\left(1/\varepsilon\right)\right).$$

Moreover, by leveraging the diminishing stepsize $\alpha_t = \dfrac{2}{2+t}$, if the extra assumption of Proposition 2.2 is satisfied, the overall complexity of solving the problem becomes

$$\underbrace{\mathcal{O}\left(\log\left(1/\varepsilon\right)\right)}_{\textit{Theorem 3.2}} \cdot \underbrace{\mathcal{O}\left(1/\varepsilon\right)}_{\textit{Proposition 2.2}} = \mathcal{O}\left(\log\left(1/\varepsilon\right)/\varepsilon\right).$$

Proof of Theorem 3.2. Using the definition of S_k in (7) and the fact that $S_0 = \lambda_0 = 1$, we can write

$$S_{k+1} = S_k + \lambda_{k+1} = \left(1 + \frac{1}{\gamma}\right) S_k.$$

By using Lemma 2.5, the corollary of Proposition 3.1, and using the sequence $S_{k+1} = \left(1+\frac{1}{\gamma}\right) S_k$ with $S_0 = 1$, we can conclude that $S_k = \left(1+\frac{1}{\gamma}\right)^k$. Therefore, we obtain

$$f(\tilde{y}_k) \leq \frac{1}{S_k}\Delta_k \leq \frac{\Delta_0}{\left(1+\frac{1}{\gamma}\right)^k} = \left(\frac{\gamma}{1+\gamma}\right)^k \Delta_0 = \left(1 - \frac{1}{1+\gamma}\right)^k \Delta_0 \leq \Delta_0 e^{-\frac{k}{\gamma+1}}.$$

Now, we just need to approximate Δ_0. Using the definition of Δ and Ψ, we can write

$$\begin{aligned}
\Delta_0 &= \max_{x\in\mathcal{V}} \Psi_0(x) = \max_{x\in\mathcal{V}} \psi^{\mu}_{y_0}(x) = \max_{x\in\mathcal{V}}\left\{\langle g(x_0), x_0 - x\rangle - \frac{\mu}{2}\|x - x_0\|^2\right\}\\
&= \max_{x\in\mathcal{V}}\left\{\langle g(x_0) - g(x^*), x_0 - x\rangle + \langle g(x^*), x_0 - x\rangle - \frac{\mu}{2}\|x - x_0\|^2\right\}\\
&\leq \langle g(x^*), x_0 - x\rangle + \max_{x\in\mathcal{V}}\left\{\langle g(x_0) - g(x^*), x_0 - x\rangle - \frac{\mu}{2}\|x - x_0\|^2\right\} \leq \langle g(x^*), x_0 - x\rangle + \frac{L^2}{2\mu}\|x^* - x_0\|^2.
\end{aligned}$$

Then, by the definition and strong convexity of the gap function f in (6), the last inequality is less than $f(x_0)$. Thus, the theorem's statement is concluded. □

We close this section by noting that the total computational time for solving the VIP (1) is the product of the convergence complexity of the considered algorithm and the complexity of the mathematical operations, e.g., projection(s) used at each iteration, which can be computationally demanding in some problems. In the next section, we see this behavior and observe the effectiveness of our proposed projection-free method compared to commonly used state-of-the-art algorithms for strongly monotone VIP.

4. Numerical Results

We demonstrate the performance of Algorithm 2 on the traffic assignment problem, an important problem studied in the transportation and operations research literature. To evaluate our performance, we compare our proposed algorithm (FW-VIP) with projected gradient descent (PGD) [30] and Nesterov's accelerated gradient method (NAG) [32], two methods commonly used for solving strongly monotone VIP in the literature. Before proceeding with the simulation results, let us first briefly explain the traffic assignment problem.

Traffic Assignment Problem (TAP): Assume N be the set of nodes, $\mathcal{E}$ be set of directed links, and W denotes the set of origin-destination (OD) pairs. We assume for all $w \in W$ there is $d_w > 0$ representing traffic demand entering from origin and exiting from destination. For each OD pair $w \in W$, the demand d_w is to be distributed among a given P_w of a paths joining w. Defining x_p that shows the flow carried by path P, we introduce the feasible path flow vector $x = \{x_p \,|\, p \in P_w,\, w \in W\}$, and the set corresponding to feasible path flow vector as

$$X = \Big\{x \,|\, \sum_{p\in P_w} x_p = d_w,\, x_p \geq 0,\, \forall p \in P_w,\, w \in W\Big\}.$$

Each collection of path flows $x \in X$ defines a collection of links flows y_{ij} with

$$y_{ij} = \sum_{w\in W} \sum_{p\in P_w} \delta_p(i,j)x_p, \quad (i,j) \in \mathcal{E}, \tag{12}$$

where $\delta_p(i,j) = 1$ if path p contains link (i,j) and zero otherwise. The vector of links flows $y = \{y_{ij} \,|\, (i,j) \in \mathcal{E}\}$ can be written as $y = Ax$ where A is the arc-chain matrix defined by (12). Similarly, we can define the set of feasible link flows as $Y = AX = \Big\{y \,|\, y = Ax,\, x \in X\Big\}$. For each link $(i,j) \in \mathcal{E}$, there is a given function $T_{ij} : Y \to \mathbb{R}$, which is the measure of the delay in link (i,j). The vector with components $T_{ij}(y)$ is $T(y)$ and due to the choice of T_{ij} it is mostly strongly monotone operator. For each $x \in X$ and corresponding $y = Ax$, the vector $T(y)$ defines the following function for each $w \in W$ and $p \in P_w$

$$\bar{T}_p(x) = \sum_{(i,j)\in\mathcal{E}} \delta_p(i,j)T_{ij}(y),$$

which shows the total travel time of path p. Then, TAP is to find $x^* \in X$ such that for all $\bar{p} \in P_w$ and $w \in W$ we have

$$\bar{T}_{\bar{p}}(x^*) = \min_{p\in P_w} \bar{T}_p(x^*) \quad \text{and} \quad x_p^* > 0. \tag{13}$$

As shown in [12], if $\bar{T}(x)$ denotes the vector with $\bar{T}_p(x)$ components, then $\bar{T}(x) = A^\top T(Ax)$ and TAP (13) can be reformulated as the following VIP:

$$\text{find } x^* \text{ such that} \quad (x - x^*)^\top \bar{T}(x^*) \geq 0, \quad \forall x \in X. \tag{14}$$

Note that $A^\top T(Ax)$ is not necessarily strongly monotone unless $A^\top A$ is inevitable (as is the case in our simulation).

We consider TAP with the same structure in [7] which can be viewed as model of a circular highway. Figure 1 illustrates the corresponding model, in which we consider five OD points, numbered 1–5, and five OD pairs: (1,4), (2,5), (3,1), (4,2), and (5,3), associated with the OD points. Each OD pair is linked with two possible paths (clockwise and counterclockwise). The links, types of links, time delays on each link, and the demand for each OD pair are given in Table 2, where $h(x) = 1 + x + x^2$ and we set $\kappa = 0.5$. Note that we assume flows cannot utilize the existing ramp that does not lead to its intended destination. Figure 2 shows the optimality gap of the corresponding VIP with parameters reported in Table 2. We note that, for a fair comparison and to demonstrate the effectiveness of the projection-free algorithm, we plot the optimality gap versus the total iterations multiplied by the total number of projections (in PGD or NAG) or LMO (in Algorithm 2). We also note that both projection and LMO operators are evaluated using the same solver, namely `OSQP` [35].

Table 2 – Parameters in TAP.

Types of links	
(1) Highway links:	17, 27, 37, 47, 57, 18, 28, 38, 48, 58.
(2) Exit ramps:	14, 24, 34, 44, 54, 12, 22, 32, 42, 52.
(3) Entrance ramps:	11, 21, 31, 41, 51, 13, 23, 33, 43, 53.
(4) Bypass links:	15, 25, 35, 45, 55, 16, 26, 36, 46, 56.
Delay on links	
(1) Delay on highway link k:	$10 \cdot h\Big[\text{flow on } k\Big] + 2\kappa \cdot h\Big[\text{flow on exit ramp from } k\Big].$
(2) Delay on exit ramp k:	$h\Big[\text{flow on } k\Big].$
(3) Delay on entrance ramp k:	$h\Big[\text{flow on } k\Big] + \kappa \cdot h\Big[\text{flow on bypass link merging with } k\Big].$
(4) Delay on bypass link k:	$h\Big[\text{flow on } k\Big].$
Demands on OD pairs	$d(1,4) = 0.1,\ d(2,5) = 0.2,\ d(3,1) = 0.3,\ d(4,2) = 0.4,\ d(5,3) = 0.5.$

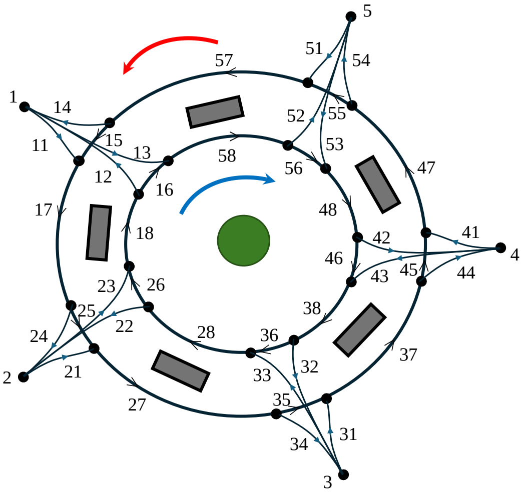

Figure 1 – TAP model (Figure 2 in [7]).

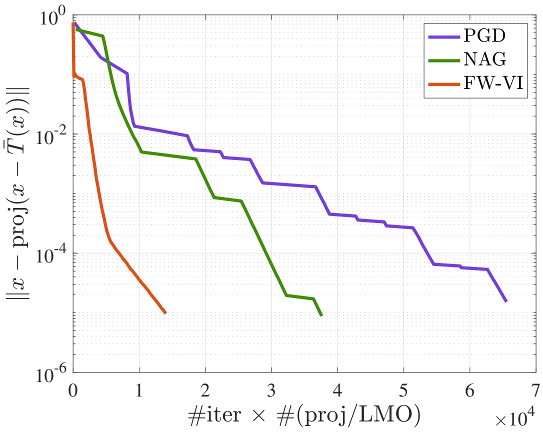

Figure 2 – Optimality gap of the VIP in the TAP.

5. Further Discussion, Limitations, and Future Directions

In this section, we provide additional insights concerning the proposed Frank-Wolfe technique for solving strongly monotone variational inequalities (1), its limitations, and potential future directions.

(i) **Further insights into the Frank-Wolfe technique for solving VIP (1):** The main motivation behind the Frank-Wolfe technique is to avoid expensive projections in each iteration and to select a more suitable descent direction compared to projected methods, which may exhibit zig-zagging behavior for certain constraint sets. This can lead to lower computational complexity and potentially faster convergence. Furthermore, as stated in Proposition 3.1, we only require the negativity of

$$N(x,y) := \min_{x_k \in \mathcal{V}} \max_{x \in \mathcal{V}} \Big\{ W(x,x_k) - \frac{\mu}{4} S_k \|x - x_k\|^2 \Big\},$$

to guarantee the convergence of Algorithm 2. While this negativity is ensured by the worst-case convergence bound in Proposition 2.2, in practice the inner Frank-Wolfe problem can be terminated as soon as $N(x,y)$ becomes negative, significantly reducing computational complexity.

(ii) **Extension to monotone variational inequalities:** Solving the VIP using a projection-free algorithm requires solving a minimax subproblem. Unfortunately, existing projection-free methods for minimax problems rely on strong convexity-concavity of the underlying function, which in our setting can be ensured by assuming strong monotonicity of the operator in the VIP. Developing a Frank-Wolfe method for minimax problems without this assumption is a key step toward extending these methods to general monotone variational inequality problems and is an important direction for future work.

(iii) **Relation to prior works and alternative assumptions:** Based on the structure of the VIP, alternative assumptions can be applied for the Frank-Wolfe algorithm in solving the inner minimax problems, replacing the requirement in Proposition 2.2 that the solution lies strictly within the domain:

- The sets $\mathcal{X}$ and $\mathcal{Y}$ are polytopes (case (II) in Theorem 1 [15]): In this case, the interior-point assumption is not required. Instead, we can use the away-step Frank-Wolfe algorithm (Algorithm 3 in [15]) and the concept of pyramidal width (Eq. 9 in [22]) instead of the border distance $\sigma_{\mathcal{X}}$, while still guaranteeing the same convergence as in Proposition 2.2.
- More recently, [10] proposed a projection-free algorithm (Algorithm 3) under a similar assumption to Proposition 2.2, without requiring the solution to lie in the interior of $\mathcal{X} \times \mathcal{Y}$. This algorithm requires $\tilde{\mathcal{O}}(1/\sqrt{\varepsilon})$ first-order and $\tilde{\mathcal{O}}(1/\varepsilon^2)$ linear optimization oracle calls to achieve ε-precision for the saddle-point problem (2). Its main limitation is implementation complexity, as it involves three nested loops per iteration.

(iv) **Limitations and future directions:**

- The primary limitation of the proposed method is the presence of a secondary loop in each iteration for solving the minimax problem. In the context of the Frank-Wolfe algorithm for minimizing a smooth convex function, two key inequalities, smoothness and convexity, are typically used. For the VIP, no explicit value function exists, making these inequalities difficult to apply. Although we can define a convex gap function $f(x)$ (6), whose minimization is equivalent to solving the VIP, this function is parametric, complicating the use of standard convex optimization inequalities. Restricting the domain to specific sets and leveraging duality may allow for a single-loop algorithm, which is an important future research direction. Adaptive or decreasing accuracy requirements for the inner loop may also help eliminate the secondary loop.

- Another limitation is the algorithm's dependence on problem-specific constants. Future extensions could involve adaptive stepsize methods [26], which approximate Lipschitz or strong monotonicity constants to accelerate convergence.
- Recent works, such as [24], establish Newton-type methods combined with Frank-Wolfe techniques for constrained self-concordant minimization, and [23] propose high-order methods for solving the VIP. Developing high-order methods for the VIP using Frank-Wolfe techniques represents another promising research direction.

References

[1] Elisabetta Allevi, Adriana Gnudi, Igor V Konnov, and Giorgia Oggioni. Evaluating the effects of environmental regulations on a closed-loop supply chain network: A variational inequality approach. *Annals of Operations Research*, 261(1):1–43, 2018.

[2] Reza Rahimi Baghbadorani, Emilio Benenati, and Sergio Grammatico. A douglas-rachford splitting method for solving monotone variational inequalities in linear-quadratic dynamic games. *preprint available at arXiv:2504.05757*, 2025.

[3] Reza Rahimi Baghbadorani, Sergio Grammatico, and Peyman Mohajerin Esfahani. Adaptive accelerated composite minimization. *preprint available at arXiv:2405.03414*, 2024.

[4] Joseph A Ball, Jerawan Chudoung, and Martin V Day. Robust optimal switching control for nonlinear systems. *SIAM Journal on Control and Optimization*, 41(3):900–931, 2002.

[5] Heinz H Bauschke, Patrick L Combettes, Heinz H Bauschke, and Patrick L Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

[6] Dimitri P Bertsekas. A class of optimal routing algorithms for communication networks. *NASA STI/Recon Technical Report N*, 81:11299, 1980.

[7] Dimitri P Bertsekas and Eli M Gafni. Projection methods for variational inequalities with application to the traffic assignment problem. In *Nondifferential and Variational Techniques in Optimization*, pages 139–159. 1982.

[8] Michiel CJ Bliemer and Piet HL Bovy. Quasi-variational inequality formulation of the multiclass dynamic traffic assignment problem. *Transportation Research Part B: Methodological*, 37(6):501–519, 2003.

[9] Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional gradient methods. *preprint available at arXiv:2211.14103*, 2022.

[10] Cheng Chen, Luo Luo, Weinan Zhang, and Yong Yu. Efficient projection-free algorithms for saddle point problems. *Advances in Neural Information Processing Systems*, 33:10799–10808, 2020.

[11] Cyrille Combettes and Sebastian Pokutta. Complexity of linear minimization and projection on some sets. *Operations Research Letters*, 49(4):565–571, 2021.

[12] Stella Dafermos. Traffic equilibrium and variational inequalities. *Transportation science*, 14(1):42–54, 1980.

[13] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

[14] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2019.

[15] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. Frank-Wolfe Algorithms for Saddle Point Problems. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 362–371. PMLR, 20–22 Apr 2017.

[16] Pontus Giselsson. Tight global linear convergence rate bounds for douglas–rachford splitting. *Journal of Fixed Point Theory and Applications*, 19(4):2241–2270, 2017.

[17] Janice Hammond. *Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms*. PhD thesis, Massachusetts Institute of Technology, 1984.

[18] Patrick Harker. A variational inequality approach for the determination of oligopolistic market equilibrium. *Mathematical Programming*, 30:105–111, 1984.

[19] Ming Hu and Masao Fukushima. Variational inequality formulation of a class of multi-leader-follower games. *Journal of optimization theory and applications*, 151:455–473, 2011.

[20] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

[21] Vladimir Kolmogorov. Solving relaxations of map-mrf problems: Combinatorial in-face frank-wolfe directions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11980–11989, 2023.

[22] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. *Advances in neural information processing systems*, 28, 2015.

[23] Tianyi Lin and Michael I Jordan. Perseus: A simple and optimal high-order method for variational inequalities. *Mathematical Programming*, pages 1–42, 2024.

[24] Deyi Liu, Volkan Cevher, and Quoc Tran-Dinh. A newton frank–wolfe method for constrained self-concordant minimization. *Journal of Global Optimization*, pages 1–27, 2022.

[25] Yura Malitsky. Projected reflected gradient methods for monotone variational inequalities. *SIAM Journal on Optimization*, 25(1):502–520, 2015.

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

[27] Yura Malitsky and Vladimir Semenov. An extragradient algorithm for monotone variational inequalities. *Cybernetics and Systems Analysis*, 50(2):271–277, 2014.

[28] Nicola Mignoni, Reza Rahimi Baghbadorani, Peyman Mohajerin Esfahani, Raffaele Carli, Mariagrazia Dotoli, and Sergio Grammatico. `monviso`: A Python package for solving monotone variational inequalities. In *European Control Conference (ECC)*, 2025. https://github.com/nicomignoni/monviso.

[29] Walaa M Moursi and Lieven Vandenberghe. Douglas–rachford splitting for the sum of a lipschitz continuous and a strongly monotone operator. *Journal of Optimization Theory and Applications*, 183:179–198, 2019.

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

[31] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, page 543, 1983.

[32] Yurii Nesterov and Laura Scrimali. Solving strongly monotone variational and quasi-variational inequalities. 2006.

[33] Georgios Pantazis, Reza Rahimi Bahbadorani, and Sergio Grammatico. Nash equilibrium seeking for a class of quadratic-bilinear wasserstein distributionally robust games. *preprint available at arXiv:2411.09636*, 2024.

[34] Gesualdo Scutari, Daniel P Palomar, Francisco Facchinei, and Jong-Shi Pang. Convex optimization, game theory, and variational inequality theory. *IEEE Signal Processing Magazine*, 27(3):35–49, 2010.

[35] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. https://github.com/osqp/osqp.

[36] Jen-Chih Yao. The generalized quasi-variational inequality problem with applications. *Journal of Mathematical Analysis and Applications*, 158(1):139–160, 1991.