2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Follower Agnostic Learning in Stackelberg Games

Chinmay Maheshwari[1], James Cheng[1], Shankar Sastry[1], Lillian Ratliff[2] and Eric Mazumdar[3]

***Abstract*— In this paper, we present an efficient algorithm to solve online Stackelberg games, featuring multiple followers, in a follower-agnostic manner. Unlike previous works, our approach works even when leader has no knowledge about the followers' utility functions, strategy space or learning algorithm. Our algorithm introduces a unique gradient estimator, leveraging specially designed strategies to probe followers. In a departure from traditional assumptions of optimal play, we model followers' responses using a convergent adaptation rule, allowing for realistic and dynamic interactions. The leader constructs the gradient estimator solely based on observations of followers' actions. We provide both non-asymptotic convergence rates to stationary points of the leader's objective and demonstrate asymptotic convergence to a *local Stackelberg equilibrium*. To validate the effectiveness of our algorithm, we use this algorithm to solve the problem of incentive design on a large-scale transportation network, showcasing its robustness even when the leader lacks access to followers' demand information.**

I. INTRODUCTION

Stackelberg games encompass a wide range of practical problems including incentive design, Bayesian persuasion, inverse optimization, bilevel optimization, cybersecurity, adversarial learning, to name a few. Stackelberg games are comprised of two type of players – *leader* and *followers*[1]. Mathematically, they are represented as follows:

$$\begin{aligned} \min_{x\in X, y\in Y} \quad & f(x,y) \\ \text{s.t.} \quad & y \in S(x) := \mathsf{SOL}(Y, G(x,\cdot))), \end{aligned} \tag{I.1}$$

where X is the leader's strategy set, $Y \subseteq \mathbb{R}^d$ is the followers' (joint) strategy set, $f : X \times Y \longrightarrow \mathbb{R}$ is the utility of the *leader*, $G : X \times Y \longrightarrow \mathbb{R}^d$ is the *game Jacobian* of *followers* and $\mathsf{SOL}(Y, G(x,\cdot))$ is a *variational inequality problem* that denotes the equilibrium response of followers, given the strategy of leader be $x \in X$. Assuming that the set $S(x)$ is singleton for every $x \in X$ (commonly referred as *lower-level singleton assumption*), (I.1) is equivalent to optimizing the following *hyper-objective*:

$$\min_{x\in X} \tilde{f}(x) := f(x, S(x)). \tag{I.2}$$

Note that in general (I.2) is non-convex optimization problem. Thus, the goal in Stackelberg games is to find a stationary point / local optima of (I.2) ([2]).

In numerous practical scenarios, it is unrealistic to presume that the leader possesses any information regarding the variational inequality problem at the lower-level, including the mapping $G(x,\cdot)$ and even their strategy set Y – information traditionally assumed in prior research on solving Stackelberg games. Thus, the key question we ask in this work is:

> **Q:** *Can we design efficient algorithms for Stackelberg games where the leader does not require any explicit knowledge of the game played between followers?*

In this work, we affirmatively answer the above question in the setting where the leader can only probe the followers with different strategies and receive estimates of their (approximated) equilibrium responses. This is in contrast to the common assumption in the literature on Stackelberg games, where it is assumed that the leader has access to an equilibrium or best-response of followers either by knowledge of the utility function of followers or through an oracle. In particular, we consider that followers are rational in the sense that they employ an adaptation/learning algorithm, which asymptotically converges to the equilibrium [3].

We propose a *two-loop* algorithm where, in the outer loop, the leader fixes its strategy (i.e., the value of x) and announces it to the followers. Between two updates of the leader's strategy, the followers employ an adaptation algorithm, for a finite number of steps, so that they converge to an *approximate* equilibrium (or best-response). Upon observing the followers' behavior, the leader constructs an approximate estimator of the gradient of the hyper-objective (I.2) and updates its strategy via gradient descent using the estimator.

We show that the proposed algorithm converges to a stationary point of (I.2) at a rate $\mathcal{O}(T^{-1/2})$. Moreover, we show that if the hyper-objective satisfies the *strict-saddle property*, i.e. the minimum eigenvalue at any saddle point is strictly negative, then the iterates asymptotically avoid saddle points (which include local maxima) and converge to a local minima of the hyper-objective (aka local Stackelberg equilibrium [2]).

We corroborate the theoretical results by conducting a simulated study of the proposed algorithm to design tolls over the Sioux Falls (South Dakota, US) transportation network. In this setup, we assume that the leader does not know the origin-destination (o-d) demand of travelers moving between different o-d pairs, which is sensitive information.

A. Related works

Learning in Stackelberg games: Learning in Stackelberg games with *finite* actions is an active area of research ([4]–[8]), where the leader has access to either a noisy or exact best response oracle. Furthermore, a dominant paradigm in

[1]CM, JC, and SS are with EECS,UC Berkeley, CA, United States.
[2]LR is with ECE, UW Seattle, WA, United States.
[3]EM is with CMS and Economics, Caltech, CA, United States.
An extended version of this article is available at [1].
[1]We shall interchangeably use the word "leader" with "upper-level" and "follower" with "lower-level".

this literature is to consider two-player games with finite strategy sets or linearly parametrized utility functions, with the exception of [2], [9]–[11]. In [2], the authors study the convergence of a two-timescale algorithm to the Stackelberg equilibrium, requiring knowledge of the Hessian of followers' utility functions for leader updates. In [9], [10], the authors require the followers to follow a specific (i.e. gradient type) learning algorithm in order to ensure convergence. Finally, in [11], the authors impose strong convexity assumption on the hyper-objective which is restrictive (as shown in [12]). In this work, we aim to design follower-agnostic learning in a general-sum Stackelberg game in continuous spaces with *no* knowledge of the followers' utility functions or learning algorithms and not imposing restrictive assumptions about convexity of hyper-objective.

Bilevel optimzation. Bilevel optimization, a subset of problem (I.1), is extensively studied in literature, resembling a Stackelberg game with a single leader and follower. Existing research on bilevel optimization pursues three main approaches. The first utilizes a value function-based approach, converting the problem into a constrained single-level optimization problem with convergence guarantees to approximate Karush-Kuhn-Tucker (KKT) points [13], [14]. However, such points may not capture locally optimal solutions [15]. Another line of research focuses on asymptotic convergence of solutions of simpler bilevel problems than (I.1) under various assumptions on the lower-level objective function structure [16]–[18]. The third line explores solving the non-convex optimization problem (I.2) using gradient descent, requiring the computation of the gradient of the solution mapping, denoted as $\nabla S(x)$. While many methods exist for approximating $\nabla S(x)$, including *Automatic Implicit Differentiation* (AID) ([19]–[23]), or *Iterative Differentiation* ([20], [24]–[26]), our work is closely related to zeroth-order methods, specifically avoiding the computation of the Hessian ([15]). Our proposed algorithm shares similarities with [15], but we eliminate the need for oracle access to a lower-level optimal solution, leveraging two-timescale stochastic approximation to analyze accumulated errors [15].

II. PROBLEM FORMULATION

Consider the following Stackelberg game

$$\begin{aligned} \min_{x\in X, y\in Y} \quad & f(x,y) \\ \text{such that} \quad & y \in S(x) := \mathsf{SOL}(Y, G(x,\cdot))), \end{aligned} \tag{SG}$$

where *(i)* $X = \mathbb{R}^d$ and $Y \subset \mathbb{R}^{d'}$ is assumed to be convex and compact set; *(ii)* $f : X \times Y \longrightarrow \mathbb{R}$ and $G : X \times Y \longrightarrow \mathbb{R}^{d'}$ are twice continuously differentiable functions; *(iii)* $\mathsf{SOL}(Y, G(x,\cdot))$ denotes the solution to variational inequality characterized by functional $G(x,\cdot)$. That is, $\mathsf{SOL}(Y, G(x,\cdot)) = \{y \in Y : \langle y' - y, G(x,y)\rangle \geq 0, \quad \forall\, y' \in Y\}$. Under mild conditions on the monotonicity of $G(x,\cdot)$, it is ensured that $S(x)$ is non-empty and convex ([27]).

In what follows, we call a continuously differentiable function $\tilde{f} : \mathbb{R}^d \longrightarrow \mathbb{R}$ to be $L-$Lipschitz if for every $x, x' \in \mathbb{R}^d$, $\|\tilde{f}(x) - \tilde{f}(x')\| \leq L\|x - x'\|$. Furthermore, we call it to be $\ell-$smooth if for every $x, x' \in \mathbb{R}^d$, $\|\nabla\tilde{f}(x) - \nabla\tilde{f}(x')\| \leq \ell\|x - x'\|$.

Next, we introduce the main assumptions on the parameters of (SG) made throughtout this paper

Assumption 1. (1) *For every $y \in Y$, the function $f(\cdot, y)$ is L_1-Lipschitz. Additionally, for every $x \in X$, the function $f(x,\cdot)$ is L_2-Lipschitz and ℓ_2-smooth.*
(2) *For every $x \in X$, the set $S(x)$ is singleton and function $S(x)$ is L_S-Lipschitz.*
(3) *The function $\tilde{f}(x) = f(x, S(x))$ is twice-continuously differentiable, $\tilde{L}$-Lipschitz and $\tilde{\ell}-$smooth.*

Assumption 1-(1) is a common assumption employed in literature to derive rates of convergence [10], [11]. Assumption 1-(2), which requires that the set $S(x)$ exists, is singleton and Lipscthiz continuous for every x, holds for strongly monotone games at lower level [28]. Furthermore, it also applies to the incentive design problem in routing games, as discussed in Section III. Assumption 1-(3) is a technical condition we impose on the hyper-objective to use Taylor's series expansion in the proof of convergence. Notably, this assumption is less restrictive than those imposed on the hyper-objective in [11]. We believe this assumption can be further relaxed, but we leave this as an interesting direction for future work.

III. MOTIVATING EXAMPLE: INCENTIVE DESIGN IN ROUTING GAMES

Consider a transportation network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ comprised of set of nodes $\mathcal{N}$ and set of edges $\mathcal{E}$, used by self-interested (infinitesimal)travelers. Each traveler is traveling between some origin-destination (o-d) pair on the network. The set of all o-d pairs be denoted by $\mathcal{Z}$. For each o-d pair $z \in \mathcal{Z}$, let $\mathcal{R}_z$ be the set of routes connecting the o-d pair z. Let D_z be the demand of travelers traveling between o-d pair $z \in \mathcal{Z}$ and y_{rz} be the flow of travelers from o-d pair $z \in \mathcal{Z}$ that choose route $r \in \mathcal{R}_z$. Naturally, $\sum_{r\in\mathcal{R}_z} y_{rz} = D_z$, for every $z \in \mathcal{Z}$. We denote the set of all feasible route flows by $Y = \prod_{z\in\mathcal{Z}} Y_z$, where $Y_z := D_z \cdot \Delta(\mathbb{R}^{|\mathcal{R}_z|})$ is a simplex. The route flow gives rise of congestion on the edges of the network. Given a route flow $y \in Y$, the resulting congestion on edges is denoted by $w(y) = (w_e(y))_{e\in\mathcal{E}}$, where

$$w_e(y) = \sum_{z\in\mathcal{Z}} \sum_{r\in\mathcal{R}_z} y_{rz}\mathbf{1}(e \in r), \quad \forall\, e \in \mathcal{E}. \tag{III.1}$$

Higher congestion leads to higher travel time on any edge. More formally, let $\ell_e(\cdot)$ be a strictly increasing smooth function which denotes the travel time of using edge $e \in \mathcal{E}$ as a function of congestion. A social planner can alter the congestion levels on the network by imposing tolls on the edges of the network which changes the preferences of travelers for different routes. Let $x_e \in \mathbb{R}$ denote the tolls imposed on edge $e \in \mathcal{E}$. [2] Under the network tolls $x = (x_e)_{e\in\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$ and route flow $y \in Y$, the overall cost

[2]Here, we allow for tolls to take negative values. Such tolling scheme can be implemented by considering revenue-refunding schemes.

experienced by travelers from o-d pair $z \in \mathcal{Z}$ taking a route $r \in \mathcal{R}_z$ is

$$c_r(y,x) = \sum_{e \in r} \ell_e(w_e(y)) + x_e. \tag{III.2}$$

Given a fixed network tolls x, the resulting congestion – *Wardrop equilibrium* – can be obtained by solving the following strictly convex optimization problem ([29])

$$S(x) = \underset{y \in Y}{\arg\min}\ \Phi(y,x) = \sum_{e \in \mathcal{E}} \int_0^{w_e(y)} \ell_e(\theta) d\theta + x_e w_e(y). \tag{III.3}$$

Under the setting presented in this section, it can be verified that the set $S(x)$ exists, is singleton, and is Lipschitz continuous mapping [28].

The goal of social planner is to minimize the overall congestion on the network while also minimizing the tolls levied on travelers. More formally, the planner's objective function is given by $f(x,y) = \sum_{e \in \mathcal{E}} w_e(y)\ell_e(w_e(y)) + \lambda \|x\|^2$, where the first term corresponds to the average congestion on the network and second term is a regularization term with parameter $\lambda > 0$, which ensures low values of tolls[3]. Thus, the problem of toll design is as follows

$$\begin{aligned} \min_{x \in \mathbb{R}^{|\mathcal{E}|}, y \in Y} \quad & f(x,y) \\ \text{s.t.} \quad & y \in S(x) = \underset{y' \in Y}{\arg\min}\ \Phi(y,x), \end{aligned} \tag{III.4}$$

which is an instantiation of (SG).

Remark 1. *In order to compute $S(x)$ in* (III.3) *the planner needs to know the set Y that requires knowledge of the demand of travelers between various o-d pairs, which is a sensitive information. In Section V, we use the proposed approach to solve* (III.4) *where the designer does not know the demand of travelers and can only observe the congestion levels $(w_e)_{e \in \mathcal{E}}$ on the network in response to the set tolls.*

IV. ALGORITHM AND ANALYSIS

In this section, we present a follower agnostic algorithm for solving (SG). Following which, we present the convergence guarantees of the proposed algorithm to a stationary point. Additionally, we show that the algorithm will eventually converge to a local optima by avoiding the saddle points and local maximum.

A. Algorithmic structure

The algorithm is based on alternatively moving towards solution to the variational inequality at lower level and descending along the upper-level objective function. Specifically, between two updates of leader (upper-level), the followers (lower level) employ an iterative adaptive rule, aimed to solve the variational inequality $\mathsf{SOL}(\cdot)$, for a fixed number of steps. Following which, the upper level iterates descend along an "approximated" gradient estimator, inspired from zeroth-order optimization ([30], [31]), evaluated at the lower-level iteration in current round.

a) Leader's strategy update: The leader's update rule is as follows:

$$x_{t+1} = x_t - \eta_t \hat{F}(x_t; \delta_t, v_t), \tag{UL}$$

where $\hat{F}(x;\delta,v)$ denotes a gradient estimator of function $\tilde{f}(\cdot) := f(\cdot, S(\cdot))$, evaluated at x with parameters δ, v. We shall describe the estimator in detail below.

b) Gradient estimator: In order to compute the gradient of $\tilde{f}(x)$, we need to compute the derivative through the solution to the variational inequality in (SG), i.e. $S(x)$, which may involve higher order gradient computations and at times is not computable in closed form due to constraints. In this work, we consider a gradient estimator inspired from [30], [31]. Specifically, we consider the following estimator

$$\hat{F}(x;\delta,v) := \frac{d}{\delta}\left(f(\hat{x}, y^{(K)}(\hat{x})) - f(x, y^{(K)}(x))\right) v, \tag{IV.1}$$

such that *(i)* $\hat{x} = x + \delta v$, where $v \in \mathcal{S}(\mathbb{R}^d) := \{z \in \mathbb{R}^d : \|z\|_2 = 1\}$ and $\delta > 0$, are referred as *perturbation* and *perturbation radius* respectively; *(ii)* K is a positive integer capturing the number of rounds of adaptation rule employed by followers between two updates of leader's strategy; *(iii)* for any $x \in X$, $k \in [K-1]$ consider a iterative solver for variational inequality denoted by H such that

$$y^{(k+1)}(x) = H_k(y^{(k)}(x); x), \quad \forall\ k \in [K-1], \tag{LL}$$

where $y^{(0)}$ is some initialization for the iterative solver of variational inequality. For example, when the lower level problem is just a convex optimization problem with objective function $g(x,\cdot)$, a typical choice of H_k is projected gradient descent, i.e. $H_k(y;x) = \mathcal{P}_Y(y - \gamma_k \nabla_y g(x,y))$, where $\mathcal{P}_Y$ denotes the orthogonal projection on Y and γ_k is the step size. Note that, in order to construct the gradient estimator in (IV.1), the leader *need not* know the exact description of update rule H_k. For most of the paper, we shall concisely denote $y^{(k)}(x)$ and $y^{(k)}(\hat{x})$ as $\tilde{y}^{(k)}$ and $y^{(k)}$ respectively.

Remark 2. *Direct application of zeroth-order gradient estimator from [30], [31] would result in following estimator*

$$\tilde{F}(x;\delta,v) = \frac{d}{\delta}\left(\tilde{f}(\hat{x}) - \tilde{f}(x)\right) v, \tag{IV.2}$$

where $\tilde{f}$ is defined in (I.2). *Observe that the gradient estimators $\hat{F}$ and $\tilde{F}$ differ because in* (IV.1) *we evaluate $f(x,\cdot)$ at $y^{(K)}(x)$ while in* (IV.2) *we evaluated it at $S(x)$ for any $x \in \mathbb{R}^d$. This induces additional bias in the gradient estimator that needs to be appropriately accounted while establishing convergence results.*

c) Algorithm: The algorithms runs for T rounds. In every round $t \in [T-1]$ the leader queries the followers with two strategies x_t and $\hat{x}_t = x_t + \delta_t v_t$ where $v_t \sim \mathsf{Unif}(\mathcal{S}(\mathbb{R}^d))$ is a vector sampled uniformly randomly from the unit sphere in $\mathbb{R}^d$ and δ_t is the *perturbation radius* (refer line **2-3** in Algorithm 1). The followers respond to the leader's

[3]Note that λ can in-general be zero, i.e. we do not require strong convexity of leader's objective function in its decision variable for our theoretical results to hold. We choose $\lambda > 0$ to impose a "soft-constraint" on the amount of tolls.

Algorithm 1: Follower Agnostic Stackelberg Optimization Algorithm

Data: Time horizon T, Initial conditions $y_0^{(0)} \in Y, \tilde{y}_0^{(0)} \in Y, x_0 \in X$, Step sizes (η_t), Perturbation radius (δ_t)

1 **for** $t = 0, 1, ...T-1$ **do**
2 Sample $v_t \sim \mathsf{Unif}(\mathcal{S}(\mathbb{R}^d))$
3 Assign $\hat{x}_t = x_t + \delta_t v_t$
4 **for** $k = 0, 1, ..., K-1$ **do**
5 Update $y_t^{(k+1)} = H_k(y_t^{(k)}; \hat{x}_t)$
6 Update $\tilde{y}_t^{(k+1)} = H_k(\tilde{y}_t^{(k)}; x_t)$
7 **end**
8 Update
$$x_{t+1} = x_t - \eta_t \frac{d}{\delta_t}\left(f(\hat{x}_t, y_t^{(K)}) - f(x_t, \tilde{y}_t^{(K)})\right) v_t$$
9 Set $y_{t+1}^{(0)} = \tilde{y}_{t+1}^{(0)} = \tilde{y}_t^{(K)}$
10 **end**

strategies by using an iterative variational inequality solver for K steps to obtain $\tilde{y}_t^{(K)}$ and $y_t^{(K)}$ respectively (refer line **4** and **7** in Algorithm 1). After observing $\tilde{y}_t^{(K)}$ and $y_t^{(K)}$, the leader computes a gradient estimator as per (IV.1). The leader updates its strategy for next time as per (UL) (refer line **8** in Algorithm 1). The followers initialize their strategies as per line **9** in Algorithm 1.

B. Convergence to stationary points

We now study the convergence properties of Algorithm 1.

Assumption 2. *For any $x, \hat{x} \in X$, the updates in* (LL) *are such that $\|y^{(K)}(x) - y^{(K)}(\hat{x})\| \leq C\|x - \hat{x}\|$, for some $C > 0$.*

Assumption 2 posits that the adaptation rule employed by followers is stable with respect to perturbations in the leader's strategy. This assumption is typically satisfied by many algorithms, including gradient-based algorithms.

Assumption 3. *Atleast one of the following holds:*

(1a) *For any $x \in X$, the iterates* (LL) *converge to equilibrium at a polynomial rate. That is, for any initial point $y^{(0)} \in Y, \|y^{(K)}(x) - S(x)\|^2 \leq CK^{-\lambda}\|y^{(0)} - S(x)\|^2$, where λ, C are positive scalars.*

(1b) *For any $x \in X$, the iterates* (LL) *converge to equilibrium at a exponential rate. That is, for any initial point $y^{(0)}$, $\|y^{(K)}(x) - S(x)\| \leq C\rho^K\|y^{(0)} - S(x)\|$, where C is a positive scalar and $\rho \in [0, 1)$.*

Remark 3. *Convergence of lower-level problem is extensively studied in literature, e.g. [32], [33], and is not the focus of this article. Assumption 3(1a) holds for gradient descent updates for convex functions that satisfy* quadratic growth *condition [34]. Meanwhile, Assumption 3(1b) holds for gradient descent on strongly convex functions.*

Theorem 1. *Let Assumption 1-3 hold. If we choose $\eta_t = \bar{\eta}(t+1)^{-1/2}d^{-1}, \delta_t = \bar{\delta}(t+1)^{-1/4}d^{-1/2}$ such that $\bar{\eta} \leq d/2\tilde{\ell}$. Then the updates $(x_t)_{t\in[T]}$ in Algorithm 1 are such that*

$$\min_{t\in[T]} \mathbb{E}\left[\|\nabla \tilde{f}(x_t)\|^2\right] \leq \tilde{O}\left(\frac{d}{\sqrt{T}} + \frac{\alpha}{1-\alpha}d^3\left(1 + \frac{1}{\sqrt{T}}\right)\right),$$

where $\alpha = CK^{-\lambda}$ if Assumption 3(1a) hold, or $\alpha = \rho^K$ if Assumption 3(1b) hold.

Intuitively, the theorem states that if we want to converge closer to a stationary point then we need to run the Algorithm 1 with larger T or smaller α (i.e. larger K). Crucially, the term αd^3 in the bound is due to error accumulation between time steps due to non-convergence of lower-level to exact solution of variational inequality $S(x)$. Owing to such precise characterization of error accumulation across time steps, our rate is informative of the *computational complexity* of solving the bi-level problem while in other contemporary work, namely [15], it resembles *iteration complexity* of the oracle. Since α is a function of K, the number of lower level iterations in every round, we can choose K to be large enough to make sure that the algorithm converges closer to the stationary point.

Corollary 1. *Let Assumption 1-2 and Assumption 3(1a) hold. Set $\eta_t = \bar{\eta}(t+1)^{-1/2}d^{-1}, \delta_t = \bar{\delta}(t+1)^{-1/4}d^{-1/2}$ such that $\bar{\eta} \leq d/2\tilde{\ell}$. Additionally, set $K \geq T^{1/2\lambda}d^{2/\lambda}$. Then the iterates of Algorithm 1 satisfy $\min_{t\in[T]} \mathbb{E}\left[\|\nabla \tilde{f}(x_t)\|^2\right] \leq \tilde{O}\left(\frac{d}{\sqrt{T}}\right)$.*

Corollary 2. *Let Assumption 1-2 and Assumption 3(1b) hold. Set $\eta_t = \bar{\eta}(t+1)^{-1/2}d^{-1}, \delta_t = \bar{\delta}(t+1)^{-1/4}d^{-1/2}$ such that $\bar{\eta} \leq d/2\tilde{\ell}$. Additionally, set $K \geq (1/|\log(\rho)|)\left((1/2)\log(T) + 2\log(d)\right)$. Then $\min_{t\in[T]} \mathbb{E}\left[\|\nabla \tilde{f}(x_t)\|^2\right] \leq \tilde{O}\left(\frac{d}{\sqrt{T}}\right)$.*

Remark 4. *We know that for non-convex smooth functions, gradient descent converges to a stationary point (at a rate of $\mathcal{O}(1/\sqrt{T})$). However, the key point of departure of* (UL) *from standard gradient descent is the presence of bias in the gradient estimator. Consequently, the key component of the proof is to bound the error in the gradient estimator* (IV.1)*. This is because the estimator can be decomposed as $\hat{F}(x_t; \delta_t, v_t) = \nabla \tilde{f}(x_t) + \mathcal{E}_t^{(1)} + \mathcal{E}_t^{(2)} + \mathcal{E}_t^{(3)}$, where $\mathcal{E}_t^{(1)} := \mathbb{E}\left[\tilde{F}(x_t; \delta_t, v_t)|x_t\right] - \nabla \tilde{f}(x_t), \mathcal{E}_t^{(2)} := \tilde{F}(x_t; \delta_t, v_t) - \mathbb{E}\left[\tilde{F}(x_t; \delta_t, v_t)|x_t\right], \mathcal{E}_t^{(3)} := \hat{F}(x_t; \delta_t, v_t) - \tilde{F}(x_t; \delta_t, v_t)$. The term $\mathcal{E}_t^{(1)}$ denotes the bias introduced due to the difference between standard zeroth-order gradient estimator, as per* (IV.2)*, and the true gradient. The term $\mathcal{E}_t^{(2)}$ denotes the randomness introduced if we were to use the standard zeroth-order gradient estimator* (IV.2)*. Finally, the term $\mathcal{E}_t^{(3)}$ denotes the bias introduced due to difference between our gradient estimator* (IV.1) *and the standard zeroth-order gradient estimator (cf. Remark 2).*

a) Proof of Theorem 1: The proof of Theorem 1 follows by noting that $\tilde{f}$ approximately decreases along the trajectory (UL) (Lemma 1). Note that the decrease is said to be "approximate" because of the bias introduced

by (IV.1) in comparison to actual gradient $\nabla\tilde{f}(\cdot)$. We then proceed to individually bound the bias terms (Lemma 2). The convergence rate follows by using the step size and perturbation radius stated in the statement of Theorem 1.

Proof of Theorem 1. From Lemma 1 we know that $\tilde{f}(\cdot)$ approximately decreases along the trajectory of (UL). That is,

$$\mathbb{E}\left[\tilde{f}(x_{t+1})\right] \leq \mathbb{E}\left[\tilde{f}(x_t)\right] - \frac{\eta_t}{2}\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] + \eta_t\mathbb{E}\left[\|\mathcal{E}_t^{(1)}\|^2\right] + \eta_t\mathbb{E}\left[\|\mathcal{E}_t^{(3)}\|^2\right] + \tilde{\ell}\eta_t^2\mathbb{E}\left[\|\mathcal{E}_t^{(2)}\|^2\right]. \tag{IV.3}$$

Using the bounds on error terms from Lemma 2, we obtain

$$\mathbb{E}\left[\tilde{f}(x_{t+1})\right] \leq \mathbb{E}\left[\tilde{f}(x_t)\right] - \frac{\eta_t}{2}\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] + \eta_t\frac{\tilde{\ell}^2\delta_t^2 d^2}{4} + \eta_t\left(\frac{d^2}{\delta_t^2}L_2^2\left(2\alpha^t e_0 + 2Cd^2\sum_{k=0}^{t-1}\alpha^{t-k}\eta_k^2 + Cd\sum_{k=0}^{t-1}\alpha^{t-k}\delta_k^2\right)\right) + 4d^2\tilde{L}^2\tilde{\ell}\eta_t^2.$$

Re-arranging the terms and adding and substracting the term $\tilde{f}(x^*) = \min_{x\in X}\tilde{f}(x)$, we obtain

$$\frac{\eta_t}{2}\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \mathbb{E}\left[\tilde{f}(x_t)\right] - \tilde{f}(x^*) + \eta_t\frac{\tilde{\ell}^2\delta_t^2 d^2}{4} + \eta_t\frac{d^2}{\delta_t^2}L_2^2\left(2\alpha^t e_0 + 2Cd^2\sum_{k=0}^{t-1}\alpha^{t-k}\eta_k^2 + C\sum_{k=0}^{t-1}\alpha^{t-k}\delta_k^2\right) - \left(\mathbb{E}\left[\tilde{f}(x_{t+1})\right] - \tilde{f}(x^*)\right) + 4d^2\tilde{L}^2\tilde{\ell}\eta_t^2.$$

Summing the previous equation over time step t we obtain

$$\sum_{t\in[T]}\eta_t\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \left(\tilde{f}(x_0) - \tilde{f}(x^*)\right) + \frac{\tilde{\ell}^2 d^2}{4}\sum_{t\in[T]}\eta_t\delta_t^2 + 2e_0 d^2 L_2^2\sum_{t\in[T]}\frac{\eta_t}{\delta_t^2}\alpha^t + \underbrace{2Cd^4L_2^2\sum_{t\in[T]}\frac{\eta_t}{\delta_t^2}\sum_{k=0}^{t-1}\alpha^{t-k}\eta_k^2}_{\text{Term E}} + \underbrace{CL_2^2d^2\sum_{t\in[T]}\frac{\eta_t}{\delta_t^2}\sum_{k=0}^{t-1}\alpha^{t-k}\delta_k^2}_{\text{Term F}} + 4d^2\tilde{L}^2\tilde{\ell}\sum_{t\in[T]}\eta_t^2 \tag{IV.4}$$

Setting $\eta_t = \bar{\eta}(t+1)^{-1/2}d^{-1}$ and $\delta_t = \bar{\delta}(t+1)^{-1/4}d^{-1/2}$, as per the statement of Theorem 1, and dividing both sides by $\sum_{t\in[T]}\eta_t$, we obtain

$$\frac{1}{\sum_{t\in[T]}\eta_t}\sum_{t\in[T]}\eta_t\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \frac{Cd}{\bar{\eta}\sqrt{T}}\left(\tilde{f}(x_0) - \tilde{f}(x^*)\right) + \frac{C\tilde{\ell}d\log(T)\bar{\delta}^2}{4\sqrt{T}} + \frac{2Cd^3L_2^2\alpha}{(1-\alpha)\bar{\eta}\sqrt{T}} + \frac{4C\tilde{L}^2\tilde{\ell}^2\bar{\eta}\log(T)d}{\sqrt{T}},$$

$$+ \frac{1}{\sum_{t\in[T]}\eta_t}\underbrace{2d^4CL_2^2\sum_{t\in[T]}\frac{\eta_t}{\delta_t^2}\sum_{k=0}^{t-1}\alpha^{t-k}\eta_k^2}_{\text{Term E}} + \frac{1}{\sum_{t\in[T]}\eta_t}\underbrace{L_2^2Cd^2\sum_{t\in[T]}\frac{\eta_t}{\delta_t^2}\sum_{k=0}^{t-1}\alpha^{t-k}\delta_k^2}_{\text{Term F}},$$

where C is a positive scalar. Next, we bound Term E + Term F as follows

$$\begin{aligned}\text{Term E } + \text{Term F} &\leq \sum_{t=1}^{T}\frac{\eta_t}{\delta_t^2}\sum_{k=0}^{t-1}\alpha^{t-k}\left(d^4\eta_k^2 + d^2\delta_k^2\right)\\ &= \frac{\bar{\eta}}{\bar{\delta}^2}\sum_{t=1}^{T}\sum_{k=0}^{t-1}\alpha^t\frac{\Theta_k}{\alpha^k} = \frac{\bar{\eta}}{\bar{\delta}^2}\sum_{k=0}^{T-1}\frac{\Theta_k}{\alpha^k}\sum_{t=k+1}^{T}\alpha^t\\ &\leq \frac{\bar{\eta}}{\bar{\delta}^2}\sum_{k=0}^{T-1}\frac{\Theta_k}{\alpha^k}\frac{\alpha^{k+1}}{1-\alpha} = \frac{\bar{\eta}}{\bar{\delta}^2}\frac{\alpha}{1-\alpha}\sum_{k=0}^{T-1}\Theta_k\\ &= \frac{\bar{\eta}}{\bar{\delta}^2}\frac{C\alpha}{1-\alpha}\left(d^2\bar{\eta}^2\log(T) + d^2\bar{\delta}^2\sqrt{T}\right),\end{aligned} \tag{IV.5}$$

where in second equality $\Theta_k := (d^4\eta_k^2 + d^2\delta_k^2)$, and we have appropriately adjusted the constant C to account for additinal constants. Thus, combining (IV.4) and (IV.5), we obtain

$$\frac{1}{\sum_{t\in[T]}\eta_t}\sum_{t\in[T]}\eta_t\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \mathcal{O}\Bigg(\frac{d}{\sqrt{T}}\left(\tilde{f}(x_0) - \tilde{f}(x^*)\right) + \frac{d\log(T)}{\sqrt{T}} + \frac{d^3\alpha}{(1-\alpha)\sqrt{T}} + \frac{4\log(T)d}{\sqrt{T}} + \frac{d}{\sqrt{T}}\frac{\alpha}{1-\alpha}\left(d^2\log(T) + d^2\sqrt{T}\right)\Bigg).$$

To conclude, we obtain

$$\min_{t\in[T]}\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \frac{1}{\sum_{t\in[T]}\eta_t}\sum_{t\in[T]}\eta_t\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] \leq \tilde{O}\left(\frac{d}{\sqrt{T}} + \frac{\alpha}{1-\alpha}d^3\left(1 + \frac{1}{\sqrt{T}}\right)\right).$$

This concludes the proof. □

Now, we formally state the Lemmas used in the proof.

Lemma 1. *If $\bar{\eta} \leq d/(2\tilde{\ell})$ then*

$$\mathbb{E}\left[\tilde{f}(x_{t+1})\right] \leq \mathbb{E}\left[\tilde{f}(x_t)\right] - \frac{\eta_t}{2}\mathbb{E}\left[\|\nabla\tilde{f}(x_t)\|^2\right] + \eta_t\mathbb{E}\left[\|\mathcal{E}_t^{(1)}\|^2\right] + \eta_t\mathbb{E}\left[\|\mathcal{E}_t^{(3)}\|^2\right] + \tilde{\ell}\eta_t^2\mathbb{E}\left[\|\mathcal{E}_t^{(2)}\|^2\right]. \tag{IV.6}$$

The proof of Lemma 1 follows in two steps: First, we use second-order Taylor series expansion of $\tilde{f}$ along the iterate values. Second, we use (UL) and complete the squares using algebraic manipulations. A detailed proof is provided in the extended version of this paper [1].

Lemma 2. *The errors $\mathbb{E}\left[\|\mathcal{E}_t^{(i)}\|^2\right]$ for $i \in \{1,2,3\}$ are bounded as follows:*

$$\mathbb{E}\left[\|\mathcal{E}_t^{(1)}\|^2\right] \leq \frac{\tilde{\ell}^2\delta_t^2 d^2}{4}, \quad \mathbb{E}\left[\|\mathcal{E}_t^{(2)}\|^2\right] \leq 4d^2\tilde{L}^2,$$

$$\mathbb{E}\left[\|\mathcal{E}_t^{(3)}\|^2\right] \leq \frac{d^2}{\delta_t^2} L_2^2 \Bigg(2\alpha^t e_0 + 2C \sum_{k=0}^{t-1} \alpha^{t-k} \eta_k^2 + C \sum_{k=0}^{t-1} \alpha^{t-k} \delta_k^2 \Bigg), \tag{IV.7}$$

where C is a scalar and $e_0 = \|y_0^{(0)} - S(x_0)\|^2$.

The stated bounds on $\mathbb{E}\left[\|\mathcal{E}_t^{(1)}\|^2\right]$ and $\mathbb{E}\left[\|\mathcal{E}_t^{(2)}\|^2\right]$ are inspired by the literature on two-point zeroth-order gradient estimators [30], [31]. We use the Lipschitz property of $f(x,\cdot)$ to bound

$$\|\mathcal{E}_t^{(3)}\|^2 \leq 2\frac{d^2}{\delta_t^2} L_2^2 \Big(\underbrace{\|y_t^{(K)} - S(\hat{x}_t)\|^2}_{\text{Term A}} + \underbrace{\|\tilde{y}_t^{(K)} - S(x_t)\|^2}_{\text{Term B}} \Big).$$

Following which, Term A and Term B are recursively bounded. A detailed proof is provided in the extended version of this paper [1].

C. Non-convergence to saddle points

In this section, we show that the updates in (UL) does not converge to a saddle point. Towards that goal, we make the following assumption that posits that the function $\tilde{f}(\cdot)$ satisfy the strict saddle property.

Assumption 4. *For any saddle point x^* of $\tilde{f}$, it holds that $\lambda_{\min}(\nabla^2 \tilde{f}(x^*)) < 0$.*

In the following theorem, we formally state the non-convergence result.

Theorem 2. *Let Assumption 1-4 hold. For $\epsilon > 0$ there exists a time T_ϵ such that for any saddle point x^* of $\tilde{f}$ it holds that $\mathbb{E}\left[\|x_t - x_*\|^2\right] \geq \epsilon, \forall\ t \geq T_\epsilon$.*

To prove Theorem 2, an asymptotic pseudo-trajectory of (UL) is constructed. We then show that the asymptotic pseudo-trajectory almost surely avoids saddle point.

a) Proof of Theorem 2: The proof follows by contradiction. Suppose there exists a saddle point x^* such that $\lim_{t\longrightarrow\infty} \mathbb{E}\left[\|x_t - x^*\|^2\right] = 0$. This implies that for any $\epsilon > 0$ there exists an integer T_ϵ such that for all $t \geq T_\epsilon$ it holds that

$$\mathbb{E}\left[\|x_{t+s} - x^*\|^2\right] \leq \epsilon/4 \quad \forall s \geq 0. \tag{IV.8}$$

Next, for any arbitrary point x_t along the trajectory (UL), we define a dynamics parametrized by $\hat{x}_t = x_t + \delta_t v_t$, as follows $z_{s+1}(\hat{x}_t) := z_s(\hat{x}_t) - \eta_{t+s}\nabla\tilde{f}(z_s(\hat{x}_t))$, where $z_0(\hat{x}_t) = \hat{x}_t$. From Lemma 3, we know that for any $\epsilon > 0$ and positive integer S there exists $\tilde{T}_{\epsilon,S}$ such that

$$\sup_{s\in[0,S]} \mathbb{E}\left[\|z_s(\hat{x}_t) - x_{t+s}\|^2\right] \leq \epsilon/4 \quad \forall\ t \geq \tilde{T}_{\epsilon,S}. \tag{IV.9}$$

Next, note that $\|z_s(\hat{x}_t) - x^*\|^2 \leq 2\|z_s(\hat{x}_t) - x_{t+s}\|^2 + 2\|x_{t+s} - x^*\|^2$. Therefore, combining (IV.8)-(IV.9), we observe that for every $t \geq \max\{T_\epsilon, T_{\epsilon,S}\}, \mathbb{E}[\|z_s(\hat{x}_t) - x^*\|^2] \leq \epsilon, \quad \forall\ s \in [0,S]$.

But from [35] we know that for gradient descent with random initialization almost surely avoids converging to saddle point [4] there exists S_ϵ such that for all $s \geq S_\epsilon$ it holds that $\|z_s(\hat{x}_t) - x^*\|^2 \geq 2\epsilon$. This establishes contradiction.

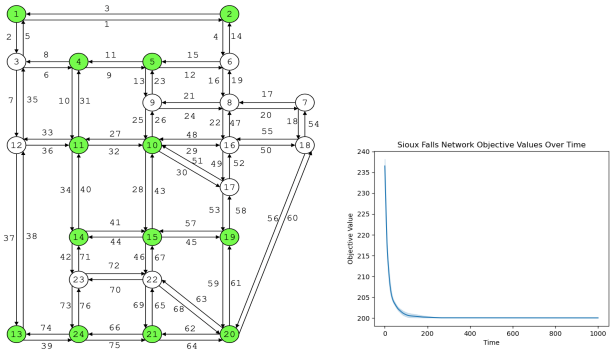
Lemma 3. *Let x_t be an arbitrary point along the trajectory (UL). Define a dynamics parametrized by $\hat{x}_t = x_t + \delta_t v_t$, such that $z_{s+1}(\hat{x}_t) := z_s(\hat{x}_t) - \eta_{t+s}\nabla\tilde{f}(z_s(\hat{x}_t))$, where $z_0(\hat{x}_t) = \hat{x}_t$ and it holds that for any positive integer L, we have*

$$\lim_{t\to\infty} \sup_{s\in[0,L]} \mathbb{E}\left[\|x_{t+s} - z_s(\hat{x}_t)\|^2\right] = 0.$$

A detailed proof of Lemma 3 is provided in the online version of this article [1].

V. NUMERICAL EXPERIMENTS

We numerically study the Algorithm 1 in the context of incentive design in routing games (described in Section III). We consider the Sioux Falls transportation network, as depicted in Figure 1(a). The latency function and network topology are inherited from `http://tinyurl.com/y4fm4nvt`. We consider a synthetic demand of $(1,2,3,2,2,1)$ units, respectively, between o-d pairs $\mathcal{Z} = ((1,20),(13,2),(20,1),(10,13),(11,20),(4,21))$.



(a) Schematic depiction of Sioux Falls transportation network network. The numbers on the edges and nodes are identifiers.

(b) The evolution of planners objective function with iterates of the algorithm. The shaded blue region denotes the confidence interval calculated over 12 runs.

Fig. 1: Simulation results.

The incentive designer can set tolls on each edge of the network. In response, unknown to the planner, the travelers alter their route selection as per a gradient rule. More formally, given a toll $x \in \mathbb{R}^{|\mathcal{E}|}$, we consider that the route choices made by the travelers are updated by descending along the gradient of the potential function $\Phi(\cdot, x)$ (cf. (III.3)). Note that, for any $x \in \mathbb{R}^{|\mathcal{E}|}, z \in \mathcal{Z}, r \in \mathcal{R}_z$, the gradient is $\frac{\partial\Phi(y,x)}{\partial y_{rz}} = \sum_{e\in\mathcal{E}} \ell_e(w_e(y)) \frac{\partial w_e(y)}{\partial y_{rz}} + x_e \frac{\partial w_e(y)}{\partial y_{rz}} \underset{(i)}{=} \sum_{e\in\mathcal{E}} \ell_e(w_e(y)) \mathbf{1}(e \in r) + x_e \mathbf{1}(e \in r) \underset{(ii)}{=} c_r(y,x)$,

[4] More specifically, we use the results from [35, Proposition 8]. Even though the results in [35, Proposition 8] hold for gradient descent update with constant step-size, we can use this result for decaying step size in our context as well. This is because the proof of [35, Proposition 8] only requires each step of the gradient update to be diffeomorphism, which holds in our setting as the step-sizes are always non-negative and decaying.

where (i) is due to (III.1) and (ii) follows from (III.2). Consequently, the gradient update takes the following form: for every $z \in \mathcal{Z}$, $y_z^{(k+1)} = \mathcal{P}_{Y_z}\left((y_{rz}^{(k)} - \gamma c_r(q^k, p))_{r\in\mathcal{R}_z}\right)$.

We simulate 12 runs of Algorithm 1 with $T = 1000$ and $K = 3$. The initial route flow vector $y_0^{(0)}$ and $\tilde{y}_0^{(0)}$ are randomly initialized. We set initial tolls uniformly randomly between $[0, 0.1]$. We set the step size $\eta_t = 6(t+1)^{-1/2}$, $\delta_t = 0.3 \cdot (t+1)^{-1/4}$, $\gamma = 0.005$ and $\lambda = 0.01$. In Figure 1(b), we show the leader's objective value as function of time iterates $t \in [T]$. We observe that all trajectories converge to same objective value even with random initializations. This shows that the convergent point is perhaps a global optimizer.

VI. CONCLUSIONS

We propose an efficient algorithm for Stackelberg games which converges to a stationary point at a rate of $\mathcal{O}(T^{-1/2})$ and asymptotically reaches a local Stackelberg equilibrium. The algorithm is designed so that the leader does not need to know any information about the game structure at lower-level and updates its strategies by only querying for the followers response to its strategy. However, in this work we assume that follower's equilibrium strategy is singleton, Lipschitz and the leader's hyper-objective is differentiable. An interesting direction of future work is to relax this assumption.

VII. ACKNOWLEDGEMENTS

This material is based on the work supported by NSF Collaborative Research: Transferable, Hierarchical, Expressive, Optimal, Robust, Interpretable NETworks (THEORINET) under grant number DMS-2031899.

REFERENCES

[1] C. Maheshwari, S. Shankar Sasty, L. Ratliff, and E. Mazumdar, "Follower agnostic methods for stackelberg games," *arXiv e-prints*, pp. arXiv–2302, 2023.

[2] T. Fiez, B. Chasnov, and L. J. Ratliff, "Convergence of learning dynamics in stackelberg games," *arXiv preprint arXiv:1906.01217*, 2019.

[3] D. Fudenberg and D. K. Levine, *The theory of learning in games*, vol. 2. MIT press, 1998.

[4] A. Blum, N. Haghtalab, and A. D. Procaccia, "Learning optimal commitment to overcome insecurity," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[5] J. Letchford, V. Conitzer, and K. Munagala, "Learning and approximating the optimal strategy to commit to," in *Algorithmic Game Theory: Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings 2*, pp. 250–262, Springer, 2009.

[6] B. Peng, W. Shen, P. Tang, and S. Zuo, "Learning optimal strategies to commit to," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 2149–2156, 2019.

[7] Y. Bai, C. Jin, H. Wang, and C. Xiong, "Sample-efficient learning of stackelberg equilibria in general-sum games," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25799–25811, 2021.

[8] P. G. Sessa, I. Bogunovic, M. Kamgarpour, and A. Krause, "Learning to play sequential games versus unknown opponents," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8971–8981, 2020.

[9] P. D. Grontas, C. Cenedese, M. Fochesato, G. Belgioioso, J. Lygeros, and F. Dörfler, "Designing optimal personalized incentive for traffic routing using big hype," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 3142–3147, IEEE, 2023.

[10] P. D. Grontas, G. Belgioioso, C. Cenedese, M. Fochesato, J. Lygeros, and F. Dörfler, "Big hype: Best intervention in games via distributed hypergradient descent," *IEEE Transactions on Automatic Control*, 2024.

[11] B. Liu, J. Li, Z. Yang, H.-T. Wai, M. Hong, Y. M. Nie, and Z. Wang, "Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence," *arXiv preprint arXiv:2110.01212*, 2021.

[12] C. Maheshwari, K. Kulkarni, M. Wu, and S. Sastry, "Adaptive incentive design with learning agents," *arXiv preprint arXiv:2405.16716*, 2024.

[13] D. Sow, K. Ji, Z. Guan, and Y. Liang, "A constrained optimization approach to bilevel optimization with multiple inner minima," *arXiv preprint arXiv:2203.01123*, 2022.

[14] M. Ye, B. Liu, S. Wright, P. Stone, and Q. Liu, "Bome! bilevel optimization made easy: A simple first-order approach," *arXiv preprint arXiv:2209.08709*, 2022.

[15] L. Chen, J. Xu, and J. Zhang, "On bilevel optimization without lower-level strong convexity," *arXiv preprint arXiv:2301.00712*, 2023.

[16] R. Liu, P. Mu, X. Yuan, S. Zeng, and J. Zhang, "A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton," in *International Conference on Machine Learning*, pp. 6305–6315, PMLR, 2020.

[17] R. Liu, X. Liu, X. Yuan, S. Zeng, and J. Zhang, "A value-function-based interior-point method for non-convex bi-level optimization," in *International Conference on Machine Learning*, pp. 6882–6892, PMLR, 2021.

[18] R. Liu, Y. Liu, S. Zeng, and J. Zhang, "Towards gradient-based bilevel optimization with non-convex followers and beyond," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8662–8675, 2021.

[19] R. Grazzi, L. Franceschi, M. Pontil, and S. Salzo, "On the iteration complexity of hypergradient computation," in *International Conference on Machine Learning*, pp. 3748–3758, PMLR, 2020.

[20] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and reverse gradient-based hyperparameter optimization," in *International Conference on Machine Learning*, pp. 1165–1173, PMLR, 2017.

[21] F. Pedregosa, "Hyperparameter optimization with approximate gradient," in *International conference on machine learning*, pp. 737–746, PMLR, 2016.

[22] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design. arxiv e-prints, art," *arXiv preprint arXiv:2010.07962*, 2020.

[23] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, "Bilevel programming for hyperparameter optimization and meta-learning," in *International Conference on Machine Learning*, pp. 1568–1577, PMLR, 2018.

[24] S. Ghadimi and M. Wang, "Approximation methods for bilevel programming," *arXiv preprint arXiv:1802.02246*, 2018.

[25] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo, "On differentiating parameterized argmin and argmax problems with application to bi-level optimization," *arXiv preprint arXiv:1607.05447*, 2016.

[26] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1723–1732, PMLR, 2019.

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

[28] S. Dafermos, "Sensitivity analysis in variational inequalities," *Mathematics of Operations Research*, vol. 13, no. 3, pp. 421–434, 1988.

[29] M. Patriksson, *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

[30] J. C. Spall, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, no. 1, pp. 109–112, 1997.

[31] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," *arXiv preprint cs/0408007*, 2004.

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

[33] S. J. Wright and B. Recht, *Optimization for data analysis*. Cambridge University Press, 2022.

[34] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, pp. 795–811, Springer, 2016.

[35] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Conference on learning theory*, pp. 1246–1257, PMLR, 2016.