

Solving Strongly Convex and Smooth Stackelberg Games Without Modeling the Follower

Yansong Li and Shuo Han

Abstract—Stackelberg games have been widely used to model interactive decision-making problems in a variety of domains such as energy systems, transportation, cybersecurity, and human-robot interaction. However, existing algorithms for solving Stackelberg games often require knowledge of the follower’s cost function or learning dynamics and may also require the follower to provide an exact best response, which can be difficult to obtain in practice. To circumvent this difficulty, we develop an algorithm that does not require knowledge of the follower’s cost function or an exact best response, making it more applicable to real-world scenarios. Specifically, our algorithm only requires the follower to provide an approximately optimal action in response to the leader’s action. The inexact best response is used in computing an approximate gradient of the leader’s objective function, with which zeroth-order bilevel optimization can be applied to obtain an optimal action for the leader. Our algorithm is proved to converge at a linear rate to a neighborhood of the optimal point when the leader’s cost function under the follower’s best response is strongly convex and smooth.

I. INTRODUCTION

Stackelberg games have recently been used as a mathematical formulation for a number of tasks in energy systems [1], [16], [18], [19], transportation [20], [23], cybersecurity [14], and human-robot interaction [11], [13], [17]. In a two-player Stackelberg game, one player, which is commonly referred to as the leader, is trying to minimize her cost assuming the other player, which is commonly referred to as the follower, is taking his optimal action after observing the leader’s action. In other words, the follower chooses a best response to the leader’s action. The Stackelberg game can also be viewed as a special case of bilevel optimization [3].

Many learning algorithms for Stackelberg games or, more generally, bilevel optimization, require knowing the follower’s cost function. For example, Fiez et al. [7] developed a learning algorithm for two-player Stackelberg games that locally converges to an optimal solution. In their algorithm, both the follower and the leader use the gradient descent algorithm, in which the leader uses an inexact gradient that depends on the first-order and second-order information on the follower’s cost function. Chen et al. [2] made the same assumption that the follower’s cost function is known and speeded up the algorithm by Fiez et al. by adding a predictive term to the follower’s learning dynamics.

When the follower’s cost function is unknown, existing algorithms require knowledge of the follower’s learning

dynamics. For example, Conn et al. [4] designed a learning algorithm for bilevel optimization that does not require first-order information on the follower’s cost function. However, they also designed the follower’s learning dynamics so that the follower is capable of providing an inexact best response. The inexact best response from the designed learning dynamics can be used by the leader to update her action. There are many ways to design the follower’s dynamics that yield an inexact best response; see [10], [22]. There are also algorithms for bilevel optimization that do not rely on designing the follower’s dynamics. However, those algorithms assume specific forms of either the follower’s optimization problem (e.g., a linear program [21]) or the leader’s problem (e.g., a least-squares problem [6]).

Assumptions of a known cost function or known learning dynamics are often made to the follower in previous works on Stackelberg games. However, these assumptions do not hold in many cases. For instance, in human-robot interaction, the follower is a human agent, whose cost function may not be known, and whose learning dynamics may not follow what is given by existing learning algorithms. Furthermore, humans do not always provide a best response, such a phenomenon is known as bounded rationality. Therefore, it is important to consider and relax these assumptions to effectively design autonomous systems that can collaborate with human agents. Existing work on Stackelberg games with a human follower typically assumes that the follower’s cost function is given or attempts to learn the cost function by, e.g., inverse reinforcement learning [8], [15]. Some results do not assume knowing the follower’s cost function but assume that the follower always provides an exact best response [11], [12].

Contribution: In this paper, we make an initial attempt to relax the current assumptions on the follower when solving two-player Stackelberg games. In particular, our problem setup makes the following assumptions:

- 1) The follower’s cost function is unknown and does not follow a specific form.
- 2) The follower’s learning dynamics are unknown and cannot be designed by us.
- 3) The follower only gives an inexact best response that is ε close to the exact best response (to be defined formally in Section III-A).

Our end result is a gradient-based learning algorithm that is guaranteed to converge to a neighborhood of the optimal point at a linear rate, where the size of the neighborhood depends on the inexactness ε of the best response oracle. Our algorithm is tested numerically for quadratic cost functions,

The authors are with the Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL 60607. Email: {yli340, hanshuo}@uic.edu. This research was supported by the Army Research Office under grant number W911NF-22-1-0034.

and the numerical simulation results are consistent with our theoretical analysis.

II. BACKGROUND: STACKELBERG GAMES

We consider a two-player Stackelberg game, where $x \in \mathbb{R}^n$ is the leader's action, and $y \in \mathbb{R}^m$ is the follower's action. The goal of the leader is to minimize her cost f_1 assuming that the follower will take the best response, i.e., the follower's action is optimal with respect to the follower's cost function given a leader's action. Mathematically, the leader attempts to solve the following bilevel optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) \triangleq f_1(x, r(x)), \quad (1)$$

where $r(x) = \arg \min_{y \in \mathbb{R}^m} f_2(x, y)$ is the best response of the follower. In this paper, we assume that the minimizer of $f_2(x, \cdot)$ is unique for each x to ensure that the best response function r is well-defined. We denote $D_j f_i(\cdot, \cdot)$ as the derivative mapping of f_i on the j -th argument, where $i = 1, 2$ and $j = 1, 2$. The derivative mapping of the best response r is denoted as Dr . The optimal solution of (1) is denoted as x^* .

The Stackelberg game formulation (1) can be used for modeling a variety of tasks. For example, in human-robot collaboration, the robot is modeled as the leader and the human as the follower. With a human follower, we typically have access to the exact zeroth-order and first-order information of the leader's cost function f_1 , but the follower's cost function f_2 and best response function r are typically unknown. We adopt these settings on f_1 , f_2 , and r together with the following assumptions:

Assumption 1. $D_1 f_1(x, \cdot)$ is $L_{f_1}^x$ -Lipschitz, and $D_2 f_1(x, \cdot)$ is $L_{f_1}^y$ -Lipschitz for any x .

Assumption 2. The best response function r is differentiable, and $\|Dr(x)\| \leq R_1$ for any x .

Assumption 3. The best response function r is R_2 -smooth, i.e., Dr is R_2 -Lipschitz continuous.

Throughout this paper, we use $\|\cdot\|$ to denote the L_2 -norm for vectors and the induced L_2 -norm for matrices.

Assumption 1 is a common assumption in bilevel optimization and Stackelberg games; see [3] and [7]. Assumptions 2 and 3 imply that the follower does not change his action rapidly when the leader's action changes.

III. PROPOSED FRAMEWORK

In this section, we give a gradient-based algorithm in Section III-A that uses an inexact gradient derived in Section III-B to solve (1). Computing the inexact gradient requires that the follower give an inexact best response to the leader's action, as defined in Section III-A.

A. Inexact Best Response

Our algorithm relies on the simple fact that the inexact best response should be close to the exact best response. A formal definition is given as follows.

Definition 1. A follower's action y is called an ε -inexact best response (ε -IBR) to a leader's action x if $\|y - r(x)\| \leq \varepsilon$.

If the follower uses gradient descent to compute an ε -IBR when the leader's action x is fixed, i.e., $y_{t+1} = y_t - \beta D_2 f_2(x, y_t)$, where β is a stepsize, then condition $\|y_t - r(x)\| < \varepsilon$ can be verified by the quantity $\|y_t - y_{t-1}\|$ if $f_2(x, \cdot)$ is strongly convex for any y .

We choose to avoid defining the inexact best response as a function $y: \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that for some constant $\varepsilon > 0$, the function y satisfies $\|y(x) - r(x)\| \leq \varepsilon$ for any x . For tasks in HRI, this would imply that the human's actions follow a function of the robot's actions. In other words, each robot's action would correspond to a specific and deterministic action of the human, thus imposing a strong assumption on the human's behavior.

Our gradient-based algorithm is given by

$$x^{k+1} = x^k - \alpha g_{x^k}, \quad (2)$$

where $k = 1, 2, \dots$ is the index of iteration. The stepsize α is a fixed constant that should be chosen small enough; the specific choice of α will be discussed in Section IV-C. The term g_{x^k} is an inexact gradient at x^k . From the decomposition of the exact gradient mapping

$$\nabla f(x) = D_1 f_1(x, r(x)) + D_2 f_1(x, r(x)) Dr(x),$$

an estimation g_x is given by

$$g_x = D_1 f_1(x, y) + D \hat{\psi}_x, \quad (3)$$

where $D_1 f_1(x, y)$ approximates $D_1 f_1(x, r(x))$, and $D \hat{\psi}_x$ approximates $D_2 f_1(x, r(x)) Dr(x)$. The follower's action y is an ε -IBR of the leader's action x . The computation of $D \hat{\psi}_x$ only requires an inexact best response of the follower. Detail will be given in Section III-B.

B. Computing an Inexact Gradient

Throughout this subsection, we denote the point where we perform gradient estimation as x_0 , i.e., we would like to use g_{x_0} as an estimate of $\nabla f(x_0)$. First, define

$$Df_1(x, y) \triangleq D_1 f_1(x, y) + D_2 f_1(x, y) Dr(x).$$

Thus, we can approximate $\nabla f(x_0)$ based on the above formula to obtain

$$\begin{aligned} \nabla f(x_0) &= Df_1(x_0, r(x_0)) \approx Df_1(x_0, y_0) \\ &= D_1 f_1(x_0, y_0) + D_2 f_1(x_0, y_0) Dr(x_0). \end{aligned}$$

The part $D_1 f_1(x_0, r(x_0))$ can be approximated by $D_1 f_1(x_0, y_0)$, where y_0 is an ε -IBR of x_0 . Similarly, to approximate the second part, define $\psi(x) \triangleq D_2 f_1(x, y_0) r(x)$. Note that the definition of $\psi(x)$ depends on (x_0, y_0) . Since $D_2 f_1(x_0, y_0) Dr(x_0) = \nabla \psi(x_0)$, we can approximate $D_2 f_1(x_0, r(x_0)) Dr(x_0)$ by a finite-difference approximation of ψ , which requires evaluating ψ at points near x_0 . However, an exact evaluation of ψ is not possible, since we do not have the exact best response r . To circumvent this issue, we

approximate $\psi(x)$ by $\hat{\psi}_x \triangleq D_2 f_1(x_0, y_0)y$, where y is an ε -IBR of x . Define each x_i as

$$x_i = x_0 + \delta v_i, \quad i = 1, 2, \dots, p, \quad (4)$$

where v_i 's is a positive basis of \mathbb{R}^n . (See [5, Section 2.1] for the definition of a positive basis.) Our goal is to approximate $D_2 f_1(x_0, r(x_0))Dr(x_0)$ from $\{(x_0, y_0), (x_1, y_1), \dots, (x_p, y_p)\}$, where each y_i is an ε -IBR of x_i . Our approximator $D\hat{\psi}_{x_0}$ for $D_2 f_1(x_0, r(x_0))Dr(x_0)$ is given by

$$\begin{aligned} D\hat{\psi}_{x_0} &= \arg \min_{D\hat{\psi}} \left\| \begin{bmatrix} \hat{\psi}_{x_0} \\ \hat{\Psi}_{x_0} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{p+1} & M \end{bmatrix} \begin{bmatrix} \hat{\psi}_{x_0} \\ D\hat{\psi} \end{bmatrix} \right\| \\ &= \arg \min_{D\hat{\psi}} \|\hat{\Psi}_{x_0} - MD\hat{\psi}\|, \end{aligned}$$

where $\hat{\Psi}_{x_0} = (\hat{\psi}_{x_1}, \dots, \hat{\psi}_{x_p})^\top$, and

$$M = \begin{bmatrix} 0 \\ \delta V \end{bmatrix}, \quad V = [v_1, \dots, v_p]^\top.$$

IV. MAIN RESULTS

In this section, we show that (2) converges at a linear rate. The difference between (2) and the standard gradient descent is that the gradient (3) used in (2) is inexact. Thus, if the inexact gradient (3) is close to the true gradient, a convergence result similar to the standard gradient descent is expected. In Section IV-A, we adopt the proof in [5, Theorem 2.13] to show that the difference between the inexact gradient computed by (3) and the true gradient is upper bounded. The upper bound relies on a function that can be further upper bounded by a function of $\|\nabla f(x)\|$ based on Assumption 4. The meaning of Assumption 4 will be discussed in detail in Section IV-B. By using the upper bound constructed in Section IV-A, we establish a similarity between (2) and the standard gradient descent and give a proof of convergence for (2) in Section IV-C.

A. Upper Bounding the Error of the Inexact Gradient

The following proposition shows that the error of the inexact gradient is upper bounded.

Proposition 1. *Suppose f_1 and f_2 satisfy Assumptions 1, 2, and 3. The inexact gradient g_x defined in (3) satisfies*

$$\|\nabla f(x) - g_x\| \leq \varphi(x) \quad (5)$$

for all x , where

$$\varphi(x) \triangleq a\varepsilon + b\|D_2 f_1(x, r(x))\| \quad (6)$$

with $a = L_{f_1}^y R_1 + L_{f_1}^x + bL_{f_1}^y$ and $b = \sqrt{p+1}(\delta^2 R_2 + \varepsilon)\|M^\dagger\|/2$.

Proof. By definition, at any point x_0

$$\begin{aligned} \|\nabla f(x_0) - g_{x_0}\| &\leq \|D_1 f_1(x_0, r(x_0)) - D_1 f_1(x_0, y_0)\| \\ &\quad + \|D_2 f_1(x_0, r(x_0))Dr(x_0) - D\hat{\psi}_{x_0}\|. \end{aligned} \quad (7)$$

The first part of the right-hand side of (7) can be bounded as

$$\begin{aligned} \|D_1 f_1(x_0, r(x_0)) - D_1 f_1(x_0, y_0)\| &\leq L_{f_1}^x \|r(x_0) - y_0\| \\ &\leq L_{f_1}^x \varepsilon. \end{aligned} \quad (8)$$

Define $\Psi_{x_0} = (\psi(x_1), \dots, \psi(x_p))^\top$, where each x_i is defined in (4). The second part of the right-hand side of (7) is bounded by

$$\begin{aligned} \|D_2 f_1(x_0, r(x_0))Dr(x_0) - D\hat{\psi}_{x_0}\| &\leq \|D_2 f_1(x_0, r(x_0))Dr(x_0) - D_2 f_1(x_0, y_0)Dr(x_0)\| \\ &\quad + \|D_2 f_1(x_0, y_0)Dr(x_0) - D\hat{\psi}_{x_0}\|. \end{aligned} \quad (9)$$

The first part of (9) can be bounded by

$$\begin{aligned} \|D_2 f_1(x_0, r(x_0))Dr(x_0) - D_2 f_1(x_0, y_0)Dr(x_0)\| &\leq \|D_2 f_1(x_0, r(x_0)) - D_2 f_1(x_0, y_0)\| \|Dr(x_0)\| \\ &\leq L_{f_1}^y R_1 \varepsilon. \end{aligned} \quad (10)$$

The second part of (9) can be bounded by

$$\begin{aligned} \|D_2 f_1(x_0, y_0)Dr(x_0) - D\hat{\psi}_{x_0}\| &= \|\nabla \psi(x_0) - D\hat{\psi}_{x_0}\| \\ &\leq \left\| \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - D\hat{\psi}_{x_0} \right\| \\ &= \left\| \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - M^\dagger \hat{\Psi}_{x_0} \right\| \\ &\leq \left\| \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - M^\dagger \Psi_{x_0} \right\| + \|M^\dagger \Psi_{x_0} - M^\dagger \hat{\Psi}_{x_0}\|. \end{aligned}$$

To bound $\left\| \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - M^\dagger \Psi_{x_0} \right\|$, define

$$h \triangleq M \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - \Psi_{x_0}. \quad (11)$$

Thus,

$$\begin{aligned} |h_i| &= |\psi(x_0) - \psi(x_i) + \langle \nabla \psi(x_0), x_i - x_0 \rangle| \\ &= \left| \int_0^1 \langle \nabla \psi(x_0) - \nabla \psi(x_0 + t(x_i - x_0)), x_i - x_0 \rangle dt \right| \\ &\leq \|x_i - x_0\| \int_0^1 \|\nabla \psi(x_0) - \nabla \psi(x_0 + t(x_i - x_0))\| dt \\ &\leq \|x_i - x_0\| \|D_2 f_1(x_0, y_0)\| \\ &\quad \cdot \int_0^1 \|Dr(x_0) - Dr(x_0 + t(x_i - x_0))\| dt \\ &\leq \|x_i - x_0\|^2 \|D_2 f_1(x_0, y_0)\| R_2 \int_0^1 t dt \\ &= \frac{\delta^2 R_2 \|D_2 f_1(x_0, y_0)\|}{2}. \end{aligned}$$

From (11),

$$\left\| \begin{bmatrix} \psi(x_0) \\ \nabla \psi(x_0) \end{bmatrix} - M^\dagger \Psi_{x_0} \right\| = \|M^\dagger h\|.$$

Also,

$$\begin{aligned}
\|M^\dagger h\| &\leq \|M^\dagger\| \|h\| \\
&\leq \sqrt{\dim(h)} \frac{\delta^2 R_2}{2} \|M^\dagger\| \|D_2 f_1(x_0, y_0)\| \\
&= \frac{\sqrt{p+1}}{2} \delta^2 R_2 \|M^\dagger\| (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon). \tag{12}
\end{aligned}$$

Finally, we can bound $\|M^\dagger \Psi_{x_0} - M^\dagger \hat{\Psi}_{x_0}\|$ by

$$\begin{aligned}
&\|M^\dagger \Psi_{x_0} - M^\dagger \hat{\Psi}_{x_0}\| \\
&\leq \|M^\dagger\| \|\Psi_{x_0} - \hat{\Psi}_{x_0}\| \\
&\leq \frac{\sqrt{p+1}}{2} \|M^\dagger\| \|D_2 f_1(x_0, y_0)\| \varepsilon \\
&\leq \frac{\sqrt{p+1}}{2} \varepsilon \|M^\dagger\| (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon). \tag{13}
\end{aligned}$$

Combine (8), (10), (12), and (13) to obtain

$$\begin{aligned}
&\|\nabla f(x_0) - g_{x_0}\| \\
&\leq L_{f_1}^x \varepsilon + L_{f_1}^y R_1 \varepsilon \\
&\quad + \frac{\sqrt{p+1}}{2} \delta^2 R_2 \|M^\dagger\| (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon) \\
&\quad + \frac{\sqrt{p+1}}{2} \varepsilon \|M^\dagger\| (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon) \\
&= L_{f_1}^x \varepsilon + L_{f_1}^y R_1 \varepsilon \\
&\quad + \frac{\sqrt{p+1}}{2} \|M^\dagger\| (\delta^2 R (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon) \\
&\quad + \varepsilon (\|D_2 f_1(x_0, r(x_0))\| + L_{f_1}^y \varepsilon)) \\
&= L_{f_1}^x \varepsilon + L_{f_1}^y R_1 \varepsilon + \frac{\sqrt{p+1}}{2} (\delta^2 R_2 + \varepsilon) \|M^\dagger\| \\
&\quad \cdot (\varepsilon L_{f_1}^y + \|D_2 f_1(x_0, r(x_0))\|).
\end{aligned}$$

Since x_0 is arbitrary, the proof is finished. \square

The value of $\|M^\dagger\|$ in the upper bound given in Proposition 1 can be calculated once a positive basis is chosen. The following corollary gives an expression of $\|M^\dagger\|$ under the positive basis that consists of the standard basis and the negative standard basis.

Corollary 1. *Choose each v_i as the positive basis defined as*

$$V = [v_1 \ v_2 \ \dots \ v_{2n}]^\top = [I_n \ -I_n].$$

In this case,

$$M = \begin{bmatrix} 0 \\ \delta I_n \\ -\delta I_n \end{bmatrix}.$$

We have $\varphi(x) = a\varepsilon + b\|D_2 f_1(x, r(x))\|$ with $a = L_{f_1}^y R_1 + L_{f_1}^x + bL_{f_1}^y$ and $b = \sqrt{4n+2}(\varepsilon/\delta + \delta R_2)/4$.

Proof. Denote by $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ the maximum and minimum singular values of a matrix, respectively. Notice $M^T M = 2\delta^2 I$. This implies $\sigma_{\min}(M) = \sqrt{2}\delta$. Use the fact $\sigma_{\max}(M^\dagger) = 1/\sigma_{\min}(M)$ to obtain $\|M^\dagger\| = \sqrt{2}/(2\delta)$. \square

B. Bounded Sensitivity

The only non-constant term in the upper bound given in (6) is $\|D_2 f_1(x_0, r(x_0))\|$. To upper bound this term, we make the following assumption.

Assumption 4 (Bounded sensitivity). There exists a constant $\kappa > 0$ such that $\|D_2 f_1(x, r(x))\| \leq \kappa \|\nabla f(x)\|$ for any x .

With Assumption 4, this term can be further bounded by $\kappa \|\nabla f(x_0)\|$, after which techniques for analyzing the standard gradient descent algorithm can be applied (see Section IV-C). This subsection will discuss the practical implications of Assumption 4.

Recall that the exact gradient of (1) is defined as

$$\nabla f(x) = D_1 f_1(x, r(x)) + D_2 f_1(x, r(x)) D r(x).$$

Under Assumption 4 and Assumption 2,

$$\begin{aligned}
&\|D_2 f_1(x, r(x))\| \\
&\leq \kappa \|D_1 f_1(x, r(x)) + D_2 f_1(x, r(x)) D r(x)\| \\
&\leq \kappa (\|D_1 f_1(x, r(x))\| + \|D_2 f_1(x, r(x))\| R_1).
\end{aligned}$$

Rearrange the above equation to obtain

$$\frac{1}{\kappa} \leq \frac{\|D_1 f_1(x, r(x))\|}{\|D_2 f_1(x, r(x))\|} + R_1 \tag{14}$$

when $\kappa \neq 0$ and $\|D_2 f_1(x, r(x))\| \neq 0$. The ratio $\|D_1 f_1(x, r(x))\| / \|D_2 f_1(x, r(x))\|$ characterizes the sensitivity of the leader's cost to the follower's action relative to the leader's action when the follower chooses the best response. Based on (14), a smaller κ implies that the leader's cost is less sensitive to the follower's action. The existence of κ implies that the sensitivity is bounded, hence the name *bounded sensitivity assumption* for Assumption 4.

Another intuitive way to understand the bounded sensitivity assumption is to consider fully collaborative Stackelberg games, where $f_1 = f_2$. In this setting, when the follower takes the best response, the leader's cost is only affected by her own action, which implies $\kappa = 0$. Formally, since $r(x)$ is the follower's best response to x , the optimality condition of the follower's optimization problem is given by $D_2 f_2(x, r(x)) = 0$. When $f_1 = f_2$, the optimality condition implies $D_2 f_1(x, r(x)) = 0$, allowing one to choose $\kappa = 0$ to satisfy Assumption 4.

C. Convergence Analysis for Strongly Convex and Smooth Cost

Under Assumption 4 and Proposition 1, we can prove formally that our algorithm converges linearly to a neighborhood of x^* by using a similar technique for analyzing standard gradient descent with strongly convex and smooth functions [9].

Theorem 1. *Suppose the follower gives an ε -IBR in every step, and the leader uses stepsize $\alpha < 1/L_f$. Also, assume that f is μ_f -strongly convex and L_f -smooth, and f_1 and f_2 satisfy Assumptions 1, 2, 3, and 4. Let $a = L_{f_1}^y R_1 + L_{f_1}^x + bL_{f_1}^y$, and $b = \sqrt{p+1}(\delta^2 R_2 + \varepsilon) \|M^\dagger\|/2$. The algorithm*

given by (2) and (3) converges to a neighborhood of f^* with the following property:

$$\limsup_{k \rightarrow \infty} (f(x_{k+1}) - f^*) \leq \frac{(2ab\kappa\varepsilon + a^2\varepsilon^2)}{2\mu_f(1 - b^2\kappa^2 - 2ab\kappa\varepsilon)}$$

if

$$b^2\kappa^2 + 2ab\kappa\varepsilon - 1 < 0. \quad (15)$$

Proof. By the Lipschitz continuity of ∇f , we have

$$\begin{aligned} f(x_{k+1}) - f(x_k) & \\ & \leq \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L_f}{2} \|x_{k+1} - x_k\|^2 \\ & = -\alpha \langle \nabla f(x_k), g(x_k) \rangle + \frac{\alpha^2 L_f}{2} \|g(x_k)\|^2 \end{aligned} \quad (16)$$

By (5),

$$\begin{aligned} \varphi(x_k)^2 & \geq \|\nabla f(x_k) - g(x_k)\|^2 \\ & = \|\nabla f(x_k)\|^2 + \|g(x_k)\|^2 - 2\langle \nabla f(x_k), g(x_k) \rangle. \end{aligned}$$

Rearrange the above inequality to obtain

$$\begin{aligned} -2\langle \nabla f(x_k), g(x_k) \rangle & \\ & \leq \varphi(x_k)^2 - \|\nabla f(x_k)\|^2 - \|g(x_k)\|^2. \end{aligned} \quad (17)$$

Substituting (17) into (16) gives

$$\begin{aligned} f(x_{k+1}) - f(x_k) & \\ & \leq -\frac{\alpha}{2} \|\nabla f(x_k)\|^2 + \left(\frac{\alpha^2 L_f}{2} - \frac{\alpha}{2} \right) \|g(x_k)\|^2 \\ & \quad + \frac{\alpha \varphi(x_k)^2}{2}. \end{aligned} \quad (18)$$

From (18),

$$\begin{aligned} f(x_{k+1}) - f(x_k) & \\ & \leq -\frac{\alpha}{2} \|\nabla f(x_k)\|^2 + \frac{\alpha \varphi(x_k)^2}{2} \\ & \leq -\frac{\alpha}{2} \|\nabla f(x_k)\|^2 + \frac{\alpha}{2} (a\varepsilon + b\kappa \|\nabla f(x_k)\|)^2 \\ & = -\frac{\alpha}{2} ((1 - b^2\kappa^2) \|\nabla f(x_k)\|^2 - a^2\varepsilon^2 \\ & \quad - 2ab\kappa\varepsilon \|\nabla f(x_k)\|). \end{aligned}$$

Since $b^2\kappa^2 + 2ab\kappa\varepsilon - 1 < 0$, if $\|\nabla f(x_k)\| \geq 1$

$$\begin{aligned} f(x_{k+1}) - f(x_k) & \\ & \leq -\frac{\alpha}{2} ((1 - b^2\kappa^2) \|\nabla f(x_k)\|^2 \\ & \quad - 2ab\kappa\varepsilon \|\nabla f(x_k)\|)^2 + \frac{\alpha a^2 \varepsilon^2}{2} \\ & \leq \mu_f \alpha (b^2\kappa^2 + 2ab\kappa\varepsilon - 1) (f(x_k) - f^*) \\ & \quad + \frac{\alpha a^2 \varepsilon^2}{2}, \end{aligned}$$

else if $\|\nabla f(x_k)\| \leq 1$, we have

$$\begin{aligned} f(x_{k+1}) - f(x_k) & \\ & \leq -\frac{\alpha}{2} (\|\nabla f(x_k)\|^2 - b^2\kappa^2 \|\nabla f(x_k)\|^2) \\ & \quad + \frac{\alpha}{2} (2ab\kappa\varepsilon + a^2\varepsilon^2) \\ & \leq \mu_f \alpha (b^2\kappa^2 - 1) (f(x_k) - f^*) \\ & \quad + \frac{\alpha}{2} (2ab\kappa\varepsilon + a^2\varepsilon^2). \end{aligned}$$

Rearrange and subtract f^* on each side of the above formula to obtain

$$\begin{aligned} f(x_{k+1}) - f^* & \leq (1 + \mu_f \alpha (b^2\kappa^2 + 2ab\kappa\varepsilon - 1)) \\ & \quad \cdot (f(x_k) - f^*) + \frac{\alpha}{2} (2ab\kappa\varepsilon + a^2\varepsilon^2). \end{aligned}$$

Note that $\alpha \leq 1/L_f \leq 1/\mu_f$. Thus, $1 > 1 + \mu_f \alpha (b^2\kappa^2 + 2ab\kappa\varepsilon - 1) \geq 0$. Take $f(x_k) - f^*$ as a Lyapunov function $V(x_k)$ finishes the proof. \square

The theorem shows that the algorithm in (2) converges to a neighborhood of x^* linearly. Because $b^2\kappa^2 + 2ab\kappa\varepsilon - 1$ is an increasing function of ε , condition (15) can always be satisfied if ε is small enough. We will discuss more on the effect of ε in Section V.

V. NUMERICAL SIMULATION

In this section, we test our algorithm when both players use a convex quadratic cost. Formally, we define $f_i: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ for $i = 1, 2$ such that

$$f_i(x, y) = \frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} P_i & Q_i \\ Q_i^\top & R_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

where

$$\begin{bmatrix} P_i & Q_i \\ Q_i^\top & R_i \end{bmatrix}$$

is positive definite. We adopt the same positive basis defined in Corollary 1 to compute the inexact gradient g_x defined by (3). Note that

$$\nabla f(x) = (P_1 - Q_1 R_2^{-1} Q_2^\top + Q_2 R_2^{-1} R_1^\top R_2^{-1} Q_2^\top) x$$

and $D_2 f_1(x, r(x)) = (P_1 - Q_1 R_2^{-1} Q_2^\top) x$. Thus, κ can be computed by

$$\kappa = \frac{\sigma_{\max}(P_1 - Q_1 R_2^{-1} Q_2^\top + Q_2 R_2^{-1} R_1^\top R_2^{-1} Q_2^\top)}{\sigma_{\min}(P_1 - Q_1 R_2^{-1} Q_2^\top)},$$

where $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ are the maximum and minimum singular values of the corresponding matrix. Also, we restrict our simulation to the case that $P_1 - Q_1 R_2^{-1} Q_2^\top + Q_2 R_2^{-1} R_1^\top R_2^{-1} Q_2^\top$ is positive definite to match the strong convexity assumption made in Section IV-C.

A. The Effect of Inexact Best Response

We chose the sampling radius δ as $\delta = 0.1$, the stepsize α as $\alpha = 0.01$, and the total number of iterations as 1000. For the problem instance and the parameters we chose, condition (15) is satisfied for $\varepsilon < 0.447$. Fig. 1 shows $\|x - x^*\|$, the distance between our iterate and the optimal point, versus the number of iterations under different choices of ε . As the plot shows, the algorithm converges to a neighborhood of x^* at a linear rate. Also, for a larger ε , the steady-state error becomes larger.

For $\varepsilon = 0.01, 0.025$, and 0.04 , the result that the steady-state error increases with ε is consistent with the upper bound

$$\frac{(2ab\kappa\varepsilon + a^2\varepsilon^2)}{2\mu_f(1 - b^2\kappa^2 - 2ab\kappa\varepsilon)} \quad (19)$$

given in Theorem 1, which is an increasing function of ε when condition (15) is satisfied. However, the numerical experiment suggests that condition (15) given in Theorem 1 may not be necessary for convergence, as shown in Fig. 1 when $\varepsilon = 0.1$ and 0.2 .

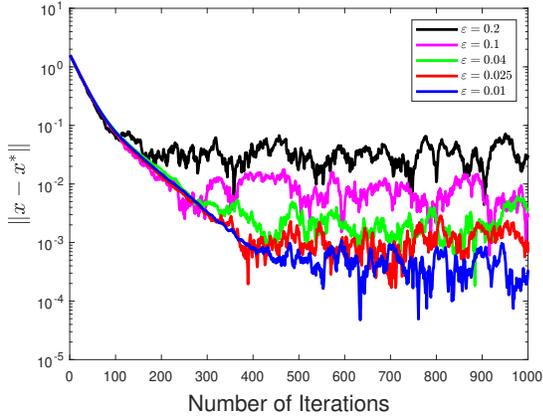


Fig. 1. Effect of ε on the convergence of (2). The steady-state error increases as ε increases.

B. Tightness of Error Bound

In this subsection, we will investigate the tightness of the theoretical error bound (19) given in Theorem 1 when condition (15) is satisfied. We calculated the gap between the theoretical error bound and the actual error, i.e.,

$$\frac{(2ab\kappa\varepsilon + a^2\varepsilon^2)}{2\mu_f(1 - b^2\kappa^2 - 2ab\kappa\varepsilon)} - (f(x_T) - f^*) \quad (20)$$

under different choices of ε , where T represents the total number of iterations.

It is easy to verify that (19) tends to 0 as ε tends to 0, which implies that the error bound is tight for $\varepsilon = 0$, i.e., when the follower gives the exact best response. However, our numerical experiment suggests that the theoretical upper bound is loose when $\varepsilon > 0$, as shown by the relationship between the gap (20) and ε given in Fig. 2. In addition, the gap (20) increases as ε increases, which suggests that the theoretical upper bound (19) in Theorem 1 becomes more conservative as ε increases.

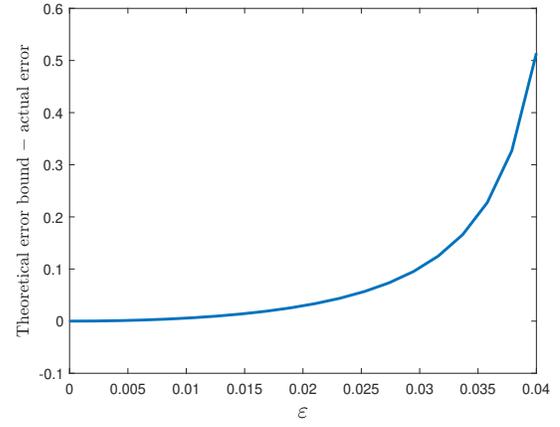


Fig. 2. Effect of ε on the tightness of the theoretical error bound when $\delta = 0.1$. The theoretical error bound becomes more conservative as ε increases.

VI. CONCLUSIONS

This paper presents an algorithm for solving Stackelberg games that relaxes several common assumptions in the literature on the follower. Unlike previous work that requires knowledge of the cost function or learning dynamics of the follower, our algorithm only requires the follower to provide an approximate best response under any action played by the leader. This is particularly relevant to interactive decision-making problems in which the follower deviates from rationality and/or has an unknown cost function, for example, when the follower is a human agent. We have shown both theoretically and numerically that our algorithm converges to a neighborhood of the optimal solution at a linear rate.

REFERENCES

- [1] Ralf Borndörfer, Bertrand Omont, Guillaume Sagnol, and Elmar Swarat. A Stackelberg Game to Optimize the Distribution of Controls in Transportation Networks. In Vikram Krishnamurthy, Qing Zhao, Minyi Huang, and Yonggang Wen, editors, *Game Theory for Networks*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 224–235, Berlin, Heidelberg, 2012. Springer.
- [2] Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A Single-Timescale Method for Stochastic Bilevel Optimization. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 2466–2488. PMLR, May 2022.
- [3] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, September 2007.
- [4] A. R. Conn and L. N. Vicente. Bilevel derivative-free optimization and its application to robust optimization. *Optimization Methods and Software*, 27(3):561–577, June 2012.
- [5] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, January 2009.
- [6] Matthias J. Ehrhardt and Lindon Roberts. Inexact Derivative-Free Optimization for Bilevel Learning. *Journal of Mathematical Imaging and Vision*, 63(5):580–600, June 2021.
- [7] Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. Implicit Learning Dynamics in Stackelberg Games: Equilibria Characterization, Convergence Analysis, and Empirical Study. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3133–3144. PMLR, November 2020.
- [8] Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

- [9] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Lojasiewicz Condition. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 9851, pages 795–811. Springer International Publishing, Cham, 2016.
- [10] Ayalew Getachew Mersha and Stephan Dempe. Direct search algorithm for bilevel programming problems. *Computational Optimization and Applications*, 49(1):1–15, May 2011.
- [11] Stefanos Nikolaidis, Jodi Forlizzi, David Hsu, Julie Shah, and Siddhartha Srinivasa. Mathematical Models of Adaptation in Human-Robot Collaboration, August 2017.
- [12] Stefanos Nikolaidis, David Hsu, and Siddhartha Srinivasa. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *The International Journal of Robotics Research*, 36(5-7):618–634, June 2017.
- [13] Stefanos Nikolaidis, Swaprava Nath, Ariel D. Procaccia, and Siddhartha Srinivasa. Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 323–331, March 2017.
- [14] Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. A Game-theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy. *ACM Comput. Surv.*, 52(4):82:1–82:28, August 2019.
- [15] Dorsa Sadigh, Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*, volume 2, pages 1–9. Ann Arbor, MI, USA, 2016.
- [16] Hichem Sedjelmaci, Makhlof Hadji, and Nirwan Ansari. Cyber Security Game for Intelligent Transportation Systems. *IEEE Network*, 33(4):216–222, July 2019.
- [17] Ran Tian, Liting Sun, Andrea Bajcsy, Masayoshi Tomizuka, and Anca D. Dragan. Safety Assurances for Human-Robot Interaction via Confidence-aware Game-theoretic Human Models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11229–11235, May 2022.
- [18] Kun Wang, Li Yuan, Toshiaki Miyazaki, Yuanfang Chen, and Yan Zhang. Jamming and Eavesdropping Defense in Green Cyber-Physical Transportation Systems Using a Stackelberg Game. *IEEE Transactions on Industrial Informatics*, 14(9):4232–4242, September 2018.
- [19] Yunpeng Wang, Walid Saad, Zhu Han, H. Vincent Poor, and Tamer Başar. A Game-Theoretic Approach to Energy Trading in the Smart Grid. *IEEE Transactions on Smart Grid*, 5(3):1439–1450, May 2014.
- [20] Hai Yang, Xiaoning Zhang, and Qiang Meng. Stackelberg games and multiple equilibrium behaviors on networks. *Transportation Research Part B: Methodological*, 41(8):841–861, October 2007.
- [21] M. Hosein Zare, Oleg A. Prokopyev, and Denis Sauré. On Bilevel Optimization with Inexact Follower. *Decision Analysis*, 17(1):74–95, March 2020.
- [22] Dali Zhang and Gui-Hua Lin. Bilevel direct search method for Leader-Follower problems and application in health insurance. *Computers & Operations Research*, 41:359–373, January 2014.
- [23] Xiaoning Zhang, H.M. Zhang, Hai-Jun Huang, Lijun Sun, and Tie-Qiao Tang. Competitive, cooperative and Stackelberg congestion pricing for multiple regions in transportation networks. *Transportmetrica*, 7(4):297–320, July 2011.