Coordinating Followers to Reach Better Equilibria: End-to-End Gradient Descent for Stackelberg Games

Kai Wang,[1] **Lily Xu,**[1] **Andrew Perrault,**[2] **Michael K. Reiter,**[3] **Milind Tambe**[1]

[1]Harvard University, [2]The Ohio State University, [3]Duke University
{kaiwang,lily_xu}@g.harvard.edu, perrault.17@osu.edu, michael.reiter@duke.edu, milind_tambe@harvard.edu

Abstract

A growing body of work in game theory extends the traditional Stackelberg game to settings with one leader and multiple followers who play a Nash equilibrium. Standard approaches for computing equilibria in these games reformulate the followers' best response as constraints in the leader's optimization problem. These reformulation approaches can sometimes be effective, but make limiting assumptions on the followers' objectives and the equilibrium reached by followers, e.g., uniqueness, optimism, or pessimism. To overcome these limitations, we run gradient descent to update the leader's strategy by differentiating through the equilibrium reached by followers. Our approach generalizes to any stochastic equilibrium selection procedure that chooses from multiple equilibria, where we compute the stochastic gradient by back-propagating through a sampled Nash equilibrium using the solution to a partial differential equation to establish the unbiasedness of the stochastic gradient. Using the unbiased gradient estimate, we implement the gradient-based approach to solve three Stackelberg problems with multiple followers. Our approach consistently outperforms existing baselines to achieve higher utility for the leader.

Introduction

Stackelberg games are commonly adopted in many real-world applications, including security (Jiang et al. 2013; Gan et al. 2020), wildlife conservation (Fang et al. 2016), and commercial decisions made by firms (Naghizadeh and Liu 2014; Aussel et al. 2020; Zhang et al. 2016). Moreover, many realistic settings involve a single leader with multiple self-interested followers such as wildlife conservation efforts with a central coordinator and a team of defenders (Gan, Elkind, and Wooldridge 2018; Gan et al. 2020); resource management in energy (Aussel et al. 2020) with suppliers, aggregators, and end users; or security problems with a central insurer and a set of vulnerable agents (Naghizadeh and Liu 2014; Johnson, Böhme, and Grossklags 2011). Solving Stackelberg games with multiple followers is challenging in general (Basilico, Coniglio, and Gatti 2017; Coniglio, Gatti, and Marchesi 2020). Previous work often reformulates the followers' best response as stationary and complementarity constraints in the leader's optimization (Shi, Zhang, and Lu 2005; Basilico et al. 2020; Basilico, Coniglio, and Gatti 2017; Coniglio, Gatti, and Marchesi 2020; Calvete and Galé 2007), casting the entire Stackelberg problem as a single optimization problem. This reformulation approach has achieved significant success in problems with linear or quadratic objectives, assuming a unique equilibrium or a specific equilibrium concept, e.g., followers' optimistic or pessimistic choice of equilibrium (Hu and Fukushima 2011; Basilico et al. 2020; Basilico, Coniglio, and Gatti 2017). The reformulation approach thoroughly exploits the structure of objectives and equilibrium to conquer the computation challenge. However, when these conditions are not met, reformulation approach may get trapped in low-quality solutions.

In this paper, we propose an end-to-end gradient descent approach to solve multi-follower Stackelberg games. Specifically, we run gradient descent by back-propagating through a sampled Nash equilibrium reached by followers to update the leader's strategy. Our approach overcomes weaknesses of reformulation approaches as (i) we decouple the leader's optimization problem from the followers', casting it as a learning problem to be solved by end-to-end gradient descent through the followers' equilibrium; and (ii) back-propagating through a sampled Nash equilibrium enables us to work with arbitrary equilibrium selection procedures and multiple equilibria.

In short, we make several contributions. First, we provide a procedure for differentiating through a Nash equilibrium assuming uniqueness (later we relax the assumption). Because each follower must simultaneously best respond to every other follower, the Karush–Kuhn–Tucker (KKT) conditions (Kuhn and Tucker 2014) for each follower must be simultaneously satisfied. We can thus differentiate through the system of KKT conditions and apply the implicit function theorem to obtain the gradient. Second, we relax the uniqueness assumption and extend our approach to an arbitrary, potentially stochastic, equilibrium selection oracle. We first show that given a stochastic equilibrium selection procedure, using optimistic or pessimistic assumptions to solve Stackelberg games with stochastic equilibria can yield payoff to the leader that is arbitrarily worse than optimal. To address the issue of multiple equilibria and stochastic equilibria, we formally characterize stochastic equilibria with a concept we call *equilibrium flow*, defined by a partial differential equation. The equilibrium flow ensures the stochas-

tic gradient computed from the sampled Nash equilibrium is unbiased, allowing us to run stochastic gradient descent to differentiate through the stochastic equilibrium. We also discuss how to compute the equilibrium flow either from KKT conditions under certain sufficient conditions or by solving the partial differential equation. This paper is the first to guarantee that the gradient computed from an arbitrary stochastic equilibrium sampled from multiple equilibria is a differentiable, unbiased sample. Third, to address the challenge that the feasibility of the leader's strategy may depend on the equilibrium reached by the followers (e.g., when a subsidy paid to the followers is conditional on their actions as in (Rotemberg 2019; Mortensen and Pissarides 2001)), we use an augmented Lagrangian method to convert the constrained optimization problem into an unconstrained one. The Lagrangian method combined with our unbiased Nash equilibrium gradient estimate enables us to run stochastic gradient descent to optimize the leader's payoff while also satisfying the equilibrium-dependent constraints.

We conduct experiments to evaluate our approach in three different multi-follower Stackelberg games: normal-form games with a leader offering subsidies to followers, Stackelberg security games with a planner coordinating multiple defenders, and cyber insurance games with an insurer and multiple customers. Across all three examples, the leader's strategy space is constrained by a budget constraint that depends on the equilibrium reached by the followers. Our gradient-based method provides a significantly higher payoff to the leader evaluated at equilibrium, compared to existing approaches which fail to optimize the leader's utility and often produce large constraint violations. These results, combined with our theoretical contributions, demonstrate the strength of our end-to-end gradient descent algorithm in solving Stackelberg games with multiple followers.

Related Work

Stackelberg models with multiple followers Multi-follower Stackelberg problems have received a lot of attention in domains with a hierarchical leader-follower structure (Nakamura 2015; Zhang et al. 2016; Liu 1998; Solis, Clempner, and Poznyak 2016; Sinha et al. 2014). Although single-follower normal-form Stackelberg games can be solved in polynomial time (Korzhyk, Conitzer, and Parr 2010; Blum et al. 2019), the problem becomes NP-hard when multiple followers are present, even when the equilibrium is assumed to be either optimistic or pessimistic (Basilico et al. 2020; Coniglio, Gatti, and Marchesi 2020). Existing approaches (Basilico et al. 2020; Aussel et al. 2020) primarily leverage the leader-follower structure in a bilevel optimization formulation (Colson, Marcotte, and Savard 2007), which can be solved by reformulating the followers' best response into non-convex stationary and complementarity constraints in the leader's optimization problem (Sinha, Soun, and Deb 2019). Various optimization techniques, including branch-and-bound (Coniglio, Gatti, and Marchesi 2020) and mixed-integer programs (Basilico et al. 2020), are adopted to solve the reformulated problems. However, these reformulation approaches highly rely on well-behaved problem structure, which may encounter large mixed integer non-linear programs when the followers have non-quadratic objectives. Additionally, these approaches mostly assume uniqueness of equilibrium or a specific equilibrium concept, e.g., optimistic or pessimistic, which may not be feasible (Gan, Elkind, and Wooldridge 2018). Previous work on the stochastic equilibrium drawn from multiple equilibria in Stackelberg problems (Lina and Jacqueline 1996) mainly focuses on the existence of an optimal solution, while our work focuses on actually solving the Stackelberg problems to identify the best action for the leader.

In contrast, our approach solves the Stackelberg problem by differentiating through the equilibrium reached by followers to run gradient descent in the leader's problem. Our approach also applies to any stochastic equilibrium drawn from multiple equilibria by establishing the unbiasedness of the gradient computed from a sampled equilibrium using a partial differential equation.

Differentiable optimization When there is only a single follower optimizing his utility function, differentiating through a Nash equilibrium reduces to the framework of differentiable optimization (Pirnay, López-Negrete, and Biegler 2012; Amos and Kolter 2017; Agrawal et al. 2019; Bai, Kolter, and Koltun 2019). When there are two followers with conflicting objectives (zero-sum), differentiating through a Nash equilibrium reduces to a differentiable minimax formulation (Ling, Fang, and Kolter 2018, 2019). Lastly, when there are multiple followers, Li et al. (2020) follow the sensitivity analysis and variational inequalities (VIs) literature (Mertikopoulos and Zhou 2019; Ui 2016; Dafermos 1988; Parise and Ozdaglar 2019) to express a unique Nash equilibrium as a fixed-point to the projection operator in VIs to differentiate through the equilibrium. Li et al. (2021) further extend the same approach to structured hierarchical games. Nonetheless, these approaches rely on the uniqueness of Nash equilibrium. In contrast, our approach generalizes to multiple equilibria.

Stackelberg Games With a Single Leader and Multiple Followers

In this paper, we consider a Stackelberg game composed of one leader and n followers. The leader first chooses a strategy $\pi \in \Pi$ that she announces, then the followers observe the leader's strategy and respond accordingly. When the leader's strategy π is determined, the followers form an n-player simultaneous game with n followers, where the i-th follower minimizes his own objective function $f_i(x_i, x_{-i}, \pi)$, which depends on his own action $x_i \in \mathcal{X}_i$, other followers' actions $x_{-i} \in \mathcal{X}_{-i}$, and the leader's strategy $\pi \in \Pi$. We assume that each strategy space is characterized by linear constraints: $\mathcal{X}_i = \{x_i \mid A_i x_i = b_i, G_i x_i \leq h_i\}$. We also assume perfect information—all the followers know other followers' utility functions and strategy spaces.

Nash Equilibria

We call $\boldsymbol{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ a Nash equilibrium if no follower has an incentive to deviate from their current strategy,

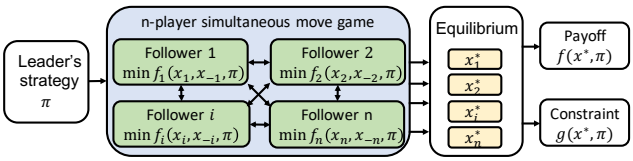


Figure 1: Given leader's strategy π, followers respond to the leader's strategy and reach a Nash equilibrium $\boldsymbol{x}^*$. The leader's payoff and the constraint depend on both the leader's strategy π and the equilibrium $\boldsymbol{x}^*$.

where we assume each follower *minimizes*[1] his objective:

$$\forall i : f_i(x_i^*, x_{-i}^*, \pi) \leq f_i(x_i, x_{-i}^*, \pi) \quad \forall x_i \in \mathcal{X}_i. \quad (1)$$

As shown in Figure 1, when the leader's strategy π is chosen and passed to an n-player game composed of all followers, we assume the followers converge to a Nash equilibrium $\boldsymbol{x}^*$.

In the first section, we assume there is a unique Nash equilibrium returned by an oracle $\boldsymbol{x}^* = \mathcal{O}(\pi)$. We later generalize to the case where there are multiple equilibria with a stochastic equilibrium selection oracle which randomly outputs an equilibrium $\boldsymbol{x} \sim \mathcal{O}(\pi)$ drawn from a distribution with probability density function $p(\cdot, \pi) : \mathcal{X} \to \mathbb{R}_{\geq 0}$.

Leader's Optimization Problem

When the leader chooses a strategy π and all the followers reach an equilibrium $\boldsymbol{x}^*$, the leader receives a payoff $f(\boldsymbol{x}^*, \pi)$ and a constraint value $g(\boldsymbol{x}^*, \pi)$. The goal of the Stackelberg leader is to choose an optimal π to maximize her utility while satisfying the constraint.

Definition 1 (Stackelberg problems with multiple followers and unique Nash equilibrium)**.** *The leader chooses a strategy π to maximize her utility function f subject to constraints g evaluated at the unique equilibrium $\boldsymbol{x}^*$ induced by an equilibrium oracle $\mathcal{O}$, i.e.,:*

$$\max_\pi f(\boldsymbol{x}^*, \pi) \quad \textit{s.t.} \quad \boldsymbol{x}^* = \mathcal{O}(\pi), \quad g(\boldsymbol{x}^*, \pi) \leq 0. \quad (2)$$

This problem is hard because the objective $f(\boldsymbol{x}^*, \pi)$ depends on the Nash equilibrium $\boldsymbol{x}^*$ reached by the followers. Moreover, notice that the feasibility constraint $g(\boldsymbol{x}^*, \pi)$ also depends on the equilibrium, which creates a complicated feasible region for the leader's strategy π.

Gradient Descent Approach

To solve the leader's optimization problem, we propose to run gradient descent to optimize the leader's objective. This requires us to compute the following gradient:

$$\frac{df(\boldsymbol{x}^*, \pi)}{d\pi} = f_\pi(\boldsymbol{x}^*, \pi) + f_{\boldsymbol{x}}(\boldsymbol{x}^*, \pi) \cdot \frac{d\boldsymbol{x}^*}{d\pi}. \quad (3)$$

The terms $f_\pi, f_{\boldsymbol{x}}$ above are easy to compute since the payoff function f is explicitly given. The main challenge is to compute $\frac{d\boldsymbol{x}^*}{d\pi}$ because it requires estimating how the Nash equilibrium $\boldsymbol{x}^*$ reached by followers responds to any change in the leader's strategy π.

[1]We use minimization formulation to align with the convention in convex optimization. In our experiments, examples of maximization problems are used, but the same approach applies.

Gradient of Unique Nash Equilibrium

In this section, we assume a unique Nash equilibrium reached by followers. Motivated by the technique proposed by Amos and Kolter (2017), we show how to differentiate through multiple KKT conditions to derive the derivative of a Nash equilibrium.

Differentiating Through KKT Conditions

Given the leader's strategy π, we express the KKT conditions of follower i with dual variables λ_i^* and ν_i^* by:

$$\begin{cases} \nabla_{x_i} f_i(x_i^*, x_{-i}^*, \pi) + G_i^\top \lambda_i^* + A_i^\top \nu_i^* = 0 \\ \text{Diag}(\lambda_i^*)(G_i x_i^* - h_i) = 0 \\ A_i x_i^* = b_i. \end{cases} \quad (4)$$

We want to estimate the impact of π on the resulting Nash equilibrium $\boldsymbol{x}^*$. Supposing the objective functions $f_i \in C^2$ are twice-differentiable, we can compute the total derivative of the the KKT system in Equation 4 written in matrix form:

$$\begin{bmatrix} \nabla^2_{x_i x_i} f_i & \nabla^2_{x_{-i} x_i} f_i & G_i^\top & A_i^\top \\ \text{Diag}(\lambda_i^*) G_i & 0 & \text{Diag}(G_i x_i^* - h_i) & 0 \\ A_i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} dx_i^* \\ dx_{-i}^* \\ d\lambda_i^* \\ d\nu_i^* \end{bmatrix}$$

$$= \begin{bmatrix} -\nabla^2_{\pi x_i} f_i d\pi - dG_i^\top \lambda_i^* - dA_i^\top \nu_i^* \\ -\text{Diag}(\lambda_i^*)(dG_i x_i^* - dh_i) \\ db_i - dA_i x_i^* \end{bmatrix}.$$

Since we assume the constraint matrices are constant, dG_i, dh_i, dA_i, db_i can be ignored. We concatenate the linear system for every follower i and move $d\pi$ to the denominator:

$$\begin{bmatrix} \nabla_{\boldsymbol{x}} F & G^\top & A^\top \\ \text{Diag}(\boldsymbol{\lambda}^*) G & \text{Diag}(G\boldsymbol{x}^* - h) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{d\boldsymbol{x}^*}{d\pi} \\ \frac{d\boldsymbol{\lambda}^*}{d\pi} \\ \frac{d\boldsymbol{\nu}^*}{d\pi} \end{bmatrix} = \begin{bmatrix} -\nabla_\pi F \\ 0 \\ 0 \end{bmatrix} \quad (5)$$

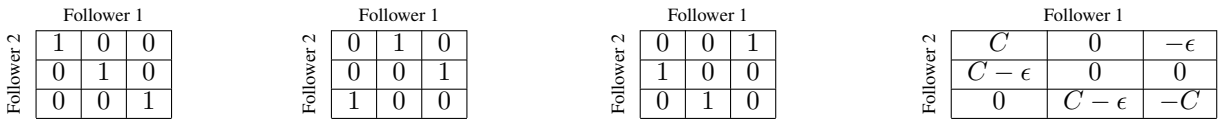
where $F = [(\nabla_{x_1} f_1)^\top, \ldots, (\nabla_{x_n} f_n)^\top]^\top$ is a column vector, and $G = \text{Diag}(G_1, G_2, \ldots, G_n), A = \text{Diag}(A_1, A_2, \ldots, A_n)$ are the diagonalized placement of a list of matrices. In particular, the KKT matrix on the left-hand side of Equation 5 matches the sensitivity analysis of Nash equilibria using variational inequalities (Facchinei, Kanzow, and Sagratella 2014; Dafermos 1988).

Proposition 1. *When the Nash equilibrium is unique and the KKT matrix in Equation 5 is invertible, the implicit function theorem holds and $\frac{d\boldsymbol{x}^*}{d\pi}$ can be uniquely determined by Equation 5.*

Proposition 1 ensures the sufficient conditions for applying Equation 5 to compute $\frac{d\boldsymbol{x}^*}{d\pi}$. Under these sufficient conditions, we can compute Equation 3 using Equation 5.

Gradient of Stochastic Equilibrium

In the previous section, we showed how to compute the gradient of a Nash equilibrium when the equilibrium is unique. However, this can be restrictive because Stackelberg games with multiple followers often have multiple equilibria that the followers can stochastically reach one. For example, both selfish routing games in the traffic setting (Roughgarden 2004) and security games with multiple defenders (Gan,



(a) Strategy 1 payoffs (b) Strategy 2 payoffs (c) Strategy 3 payoffs (d) Leader payoffs

Figure 2: Payoff matrices from Theorem 1 where the leader has 3 strategies. Follower payoffs for each strategy in (a)–(c) where both followers receive the same payoff; leader payoffs in (d).

Elkind, and Wooldridge 2018) can have multiple equilibria that are reached in multiple independent runs.

In this section, we first demonstrate the importance of stochastic equilibrium by showing that optimizing over optimistic or pessimistic equilibrium could lead to arbitrarily bad leader's payoff under the stochastic setting. Second, we generalize our gradient computation to the case with multiple equilibria, allowing the equilibrium oracle $\mathcal{O}$ to stochastically return a sample equilibrium from a distribution of multiple equilibria. Lastly, we discuss how to compute the gradient of different types of equilibria and its limitation.

Importance of Stochastic Equilibrium

When the equilibrium oracle is stochastic, our Stackelberg problem becomes a stochastic optimization problem:

Definition 2 (Stackelberg problems with multiple followers and stochastic Nash equilibria)**.** *The leader chooses a strategy π to optimize her expected utility and satisfy the constraints in expectation under a given stochastic equilibrium oracle $\mathcal{O}$:*

$$\max_{\pi} \mathop{\mathbb{E}}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f(\boldsymbol{x}^*, \pi) \quad \textit{s.t.} \mathop{\mathbb{E}}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} g(\boldsymbol{x}^*, \pi) \leq 0. \quad (6)$$

In particular, we show that if we ignore the stochasticity of equilibria by simply assuming optimistic or pessimistic equilibria, the leader's expected payoff can be arbitrarily worse than the optimal one.

Theorem 1. *Assuming the followers stochastically reach a Nash equilibrium drawn from a distribution over all equilibria, solving a Stackelberg game under the assumptions of optimistic or pessimistic equilibrium can give the leader expected payoff that is arbitrarily worse than the optimal one.*

Proof. We consider a Stackelberg game with one leader and two followers (row and column player) with no constraint. The leader can choose 3 different strategies, each corresponding to a payoff matrix in Figure 2(a)–2(c), where both followers receive the same payoff in the entry when they choose the corresponding row and column. In each payoff matrix, there are three pure Nash equilibria; we assume the followers reach any of them uniformly at random. After the followers reach a Nash equilibrium, the leader receives the corresponding entry in the payoff matrix in Figure 2(d).

Under the optimistic assumption, the leader would choose strategy 1, expecting followers to break the tie in favor of the leader, yielding payoff C. Instead, the three followers select a Nash equilibria uniformly at random, yielding expected payoff $\frac{C+0-C}{3} = 0$. Under the pessimistic assumption, the leader chooses strategy 2, anticipating and receiving an expected payoff of zero. Under the correct stochastic assumption, she chooses strategy 3 with expected payoff $\frac{C-\epsilon+C-\epsilon-\epsilon}{3} = \frac{2}{3}C - \epsilon \gg 0$, which can be arbitrarily higher than the optimistic or pessimistic payoff when $C \to \infty$. □

Theorem 1 justifies why we need to work on stochastic equilibrium when the equilibrium is drawn stochastically in Definition 2. In the following section, we show how to apply gradient descent to optimize the leader's payoff by differentiating through followers' equilibria with a stochastic oracle.

Equilibrium Flow and Unbiased Gradient Estimate

Our goal is to compute the gradient of the objective in Equation 6: $\frac{d}{d\pi} \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f(\boldsymbol{x}^*, \pi)$. However, since the distribution of the oracle $\mathcal{O}(\pi)$ can also depend on π, we cannot easily exchange the gradient operator into the expectation.

To address the dependency of the oracle $\mathcal{O}(\pi)$ on π, we use $p(\boldsymbol{x}, \pi)$ to represent the probability density function of the oracle $\boldsymbol{x} \sim \mathcal{O}(\pi)$ for every π. We want to study how the oracle distribution changes as the leader's strategy π changes, which we denote by *equilibrium flow* as defined by the following partial differential equation:

Definition 3 (Equilibrium Flow)**.** *We call $v(\boldsymbol{x}, \pi)$ the equilibrium flow of the oracle $\mathcal{O}$ with probability density function $p(\boldsymbol{x}, \pi)$ if $v(\boldsymbol{x}, \pi)$ satisfies the following equation:*

$$\frac{\partial}{\partial \pi} p(\boldsymbol{x}, \pi) = -\nabla_{\boldsymbol{x}} \cdot (p(\boldsymbol{x}, \pi) v(\boldsymbol{x}, \pi)). \quad (7)$$

This differential equation is similar to many differential equations of various conservation laws, where $v(\boldsymbol{x}, \pi)$ serves as a velocity term to characterize the movement of equilibria. In the following theorem, we use the equilibrium flow $v(\boldsymbol{x}, \pi)$ to address the dependency of $\mathcal{O}(\pi)$ on π.

Theorem 2. *If $v(\boldsymbol{x}^*, \pi)$ is the equilibrium flow of the stochastic equilibrium oracle $\mathcal{O}(\pi)$, we have:*

$$\begin{aligned} &\frac{d}{d\pi} \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f(\boldsymbol{x}^*, \pi) \\ &= \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} \left[f_\pi(\boldsymbol{x}^*, \pi) + f_{\boldsymbol{x}}(\boldsymbol{x}^*, \pi) \cdot v(\boldsymbol{x}^*, \pi) \right]. \end{aligned} \quad (8)$$

Proof sketch. To compute the derivative on the left-hand side, we can expand the expectation by:

$$\begin{aligned} &\frac{d}{d\pi} \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f(\boldsymbol{x}^*, \pi) = \frac{d}{d\pi} \int f(\boldsymbol{x}, \pi) p(\boldsymbol{x}, \pi) d\boldsymbol{x} \\ &= \int p(\boldsymbol{x}, \pi) \frac{\partial}{\partial \pi} f(\boldsymbol{x}, \pi) + f(\boldsymbol{x}, \pi) \frac{\partial}{\partial \pi} p(\boldsymbol{x}, \pi) d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f_\pi(\boldsymbol{x}^*, \pi) + \int f(\boldsymbol{x}, \pi) \frac{\partial}{\partial \pi} p(\boldsymbol{x}, \pi) d\boldsymbol{x}. \end{aligned} \quad (9)$$

We substitute the term $\frac{\partial}{\partial \pi} p = -\nabla_{\boldsymbol{x}} \cdot (p \cdot v)$ by the definition of equilibrium flow, and apply integration by parts and Stokes' theorem[2] to the right-hand side of Equation 9 to get Equation 8. More details can be found in the appendix. □

Theorem 2 extends the derivative of Nash equilibrium to the case of stochastic equilibrium randomly drawn from multiple equilibria. Specifically, Equation 9 offers an efficient unbiased gradient estimate by sampling an equilibrium from the stochastic oracle to compute the right-hand side of Equation 9. Theorem 2 also matches to Equation 3, where the role of equilibrium flow $v(\boldsymbol{x}^*, \pi)$ coincides with the role of $\frac{d\boldsymbol{x}^*}{d\pi}$ in Equation 3.

How to Determine Equilibrium Flow

The only remaining question is how to determine the equilibrium flow. Given the leader's strategy π, there are two types of equilibria: (i) isolated equilibria and (ii) non-isolated equilibria. We first show that the solution to Equation 5 matches the equilibrium flow for every equilibrium in case (i) when the probability of sampling the equilibrium is locally fixed.

Theorem 3. *Given the leader's strategy π and a sampled equilibrium $\boldsymbol{x}$, if (1) the KKT matrix at $(\boldsymbol{x}, \pi)$ is invertible and (2) $\boldsymbol{x}$ is sampled with a fixed probability locally, the solution to Equation 5 is a homogeneous solution to Equation 7 and matches the equilibrium flow $v(\pi, \boldsymbol{x})$ locally.*

Theorem 3 ensures that when the sampled equilibrium behaves like a unique equilibrium locally, the solution to Equation 5 matches the equilibrium flow of the sampled equilibrium. In particular, Theorem 3 does not require all equilibria are isolated; it works as long as the sampled equilibrium satisfies the sufficient conditions. In contrast, the study in multiple equilibria requires global isolation for the analysis to work. Together with Theorem 2, we can use the solution to Equation 5 as an unbiased equilibrium gradient estimate and run stochastic gradient descent accordingly.

Lastly, when the sufficient conditions in Theorem 3 are not satisfied, e.g., the KKT matrix becomes singular for any non-isolated equilibrium, the solution to Equation 5 does not match the equilibrium flow $v(\boldsymbol{x}, \pi)$. In this case, to compute the equilibrium flow correctly, we rely on solving the partial differential equation in Equation 7. If the probability density function $p(\boldsymbol{x}, \pi)$ is explicitly given, we can directly solve Equation 7 to derive the equilibrium flow. If the probability density function $p(\boldsymbol{x}, \pi)$ is not given, we can use the empirical equilibrium distribution $p'(\boldsymbol{x}, \pi)$ constructed from the historical equilibrium samples of the oracle instead.

In practice, we hypothesize that even if the equilibria are not isolated and the corresponding KKT matrices are singular, solving degenerated version of Equation 5 still serves as a good approximation to the equilibrium flow. Therefore, we still use the solution to Equation 5 as an approximate of the equilibrium flow in the following sections and algorithms.

[2]The analysis of integration by parts and Stokes' theorem applies to both Riemann and Lebesgue integral. Lebesgue integral is needed when the set of equilibria forms a measure-zero set.

Algorithm 1: Augmented Lagrangian Method

Initialization: $\pi = \pi_{\text{init}}$, learning rate γ, multipliers $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$, slack variable $\boldsymbol{s} \geq \mathbf{0}$, $K = 100$
for *iteration in* $\{1, 2, \dots\}$ **do**
  Define the objective to be Lagrangian $\mathcal{L}(\pi, \boldsymbol{s}; \boldsymbol{\lambda})$ defined in Equation 10
  Compute a sampled gradient of $\mathcal{L}$ by sampling $\boldsymbol{x}^* \sim \mathcal{O}(\pi)$. Compute $\frac{d\boldsymbol{x}^*}{d\pi}$ by Equation 5
  Update $\pi = \pi - \gamma(\frac{\partial \mathcal{L}}{\partial \pi} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^*} \frac{d\boldsymbol{x}^*}{d\pi})$, $\boldsymbol{s} = \max\{\boldsymbol{s} - \gamma \frac{\partial \mathcal{L}}{\partial \boldsymbol{s}}, \mathbf{0}\}$
  if *iteration is a multiple of* K **then**
    Update $\boldsymbol{\lambda} = \boldsymbol{\lambda} - \mu(g(\boldsymbol{x}^*, \pi) + \boldsymbol{s})$
Return: leader's strategy π

Gradient-Based Algorithm and Augmented Lagrangian Method

To solve both the optimization problems in Definition 1 and Definition 2, we implement our algorithm with (i) stochastic gradient descent with unbiased gradient access, and (ii) augmented Lagrangian method to handle the equilibrium-dependent constraints. We use the relaxation algorithm (Uryasev and Rubinstein 1994) as our equilibrium oracle $\mathcal{O}$. The relaxation algorithm is a classic equilibrium finding algorithm that iteratively updates agents' strategies by best responding to other agents' strategies until convergence with guarantees (Krawczyk and Uryasev 2000).

Since the leader's strategy π is constrained by the followers' response, we adopt an augmented Lagrangian method (Bertsekas 2014) to convert the constrained problem to an unconstrained one with a Lagrangian objective. We introduce a slack variable $\boldsymbol{s} \geq \mathbf{0}$ to convert inequality constraints into equality constraints $\mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)}\, g(\boldsymbol{x}^*, \pi) + \boldsymbol{s} = \mathbf{0}$. Thus, the penalized Lagrangian can be written as:

$$\mathcal{L}(\pi, \boldsymbol{s}; \boldsymbol{\lambda}) = -\mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)} f(\boldsymbol{x}^*, \pi) + \boldsymbol{\lambda}^\top (\mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)}\, g(\boldsymbol{x}^*, \pi) + \boldsymbol{s}) + \frac{\mu}{2} \left\| \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{O}(\pi)}\, g(\boldsymbol{x}^*, \pi) + \boldsymbol{s} \right\|^2. \quad (10)$$

We run gradient descent on the minimization problem of the penalized Lagrangian $\mathcal{L}(\pi, \boldsymbol{s}; \boldsymbol{\lambda})$ and update the Lagrangian multipliers $\boldsymbol{\lambda}$ every fixed number of iterations, as described in Algorithm 1. The stochastic Stackelberg problem with multiple followers can be solved by running stochastic gradient descent with augmented Lagrangian methods, where Theorem 2 ensures the unbiasedness of the stochastic gradient estimate under the conditions in Theorem 3.

Example Applications

We briefly describe three different Stackelberg games with one leader and multiple self-interested followers. Specifically, normal-form games with risk penalty has a unique Nash equilibrium, while other examples can have multiple.

Coordination in Normal-Form Games

A normal-form game (NFG) is composed of n follower players each with a payoff matrix $U_i \in \mathbb{R}^{m_1 \times \cdots \times m_n}$ for all

$i \in [n]$, where the i-th player has m_i available pure strategies. The set of all feasible mixed strategies of player i is $x_i \in \mathcal{X}_i = \{x \in [0,1]^{m_i} \mid \mathbf{1}^\top x = 1\}$. On the other hand, the leader can offer non-negative subsidies $\pi_i \in \mathbb{R}_{\geq 0}^{m_1 \times \cdots \times m_n}$ to each player i to reward specific combinations of pure strategies. The subsidy scheme is used to control the payoff matrix and incentivize the players to change their strategies.

Once the subsidy scheme π is determined, each player i chooses a strategy x_i and receives the expected payoff $U_i(\boldsymbol{x})$ and subsidy $\pi_i(\boldsymbol{x})$, subtracting a penalty term $H(x_i) = \sum_j x_{ij} \log x_{ij}$, the Gibbs entropy of the chosen strategy $\boldsymbol{x}_i$ to represent the risk aversion of player i. Since the followers' objectives are concave, the risk aversion model yields a unique Nash equilibrium, which is known to be quantal response equilibrium (QRE) (McKelvey and Palfrey 1995; Ling, Fang, and Kolter 2018). Lastly, the leader's payoff is given by the social welfare across all players, which is the summation of the expected payoffs without subsidies: $\sum_{i \in [n]} U_i(\boldsymbol{x})$. The subsidy scheme is subject to a budget constraint B on the total subsidy paid to all players.

Security Games with Multiple Defenders

Stackelberg security games (SSGs) model a defender protecting a set of targets T from being attacked. We consider a scenario with a leader coordinator and n non-cooperative follower defenders each patrolling a subset $T_i \subseteq T$ of the targets (Gan, Elkind, and Wooldridge 2018). Each defender i can determine the patrol effort spent on protecting the designated targets. We use $0 \leq x_{i,t} \leq 1$ to denote the effort spent on target $t \in T_i$ and the total effort is upper bounded by b_i. Defender i only receives a penalty $U_{i,t} < 0$ when target $t \in T_i$ in her protected region is attacked but unprotected by all defenders, and 0 otherwise.

Because the defenders are independent, the patrol strategies $\boldsymbol{x}$ can overlap, leading to a multiplicative unprotected probability $\prod_i (1 - x_{i,t})$ of target t. Given the unprotected probabilities, attacks occur under a distribution $\boldsymbol{p} \in \mathbb{R}^{|T|}$, where the distribution $\boldsymbol{p}$ is a function of the unprotected probabilities defined by a quantal response model. To encourage collaboration, the leader coordinator can selectively provide reimbursement $\pi_{i,t} \geq 0$ to alleviate defender i's loss when target t is attacked but unprotected, which encourages the defender to focus on protecting specific regions, reducing wasted effort on overlapping patrols. The leader's goal is to protect all targets, where the leader's objective is the total return across over all targets $\sum_{t \in T} U_t p_t \prod_i (1 - x_{i,t})$. Lastly, the reimbursement scheme π must satisfy a budget constraint B on the total paid reimbursement.

Cyber Insurance Games With Multiple Customers

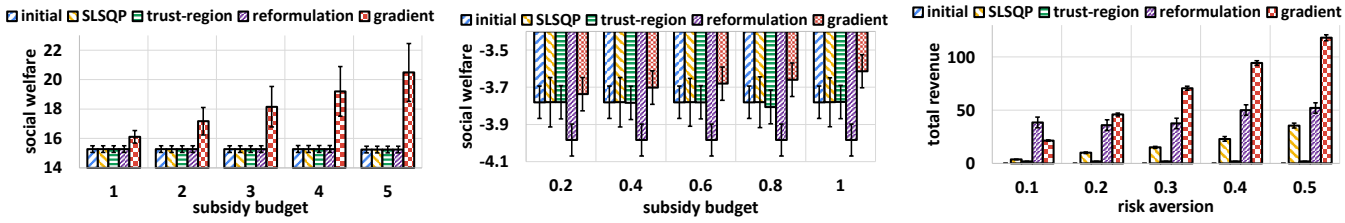
We adopt the cyber insurance model proposed by Naghizadeh et al. (2014) and Johnson et al. (2011) to study how agents in an interconnected cyber security network make decisions, where agents' decisions jointly affect each other's risk. There are n agents (followers) facing malicious cyberattacks. Each agent i can deploy an effort of protection $x_i \in \mathbb{R}_{\geq 0}$ to his computer system, where investing in protection incurs a linear cost $c_i x_i$. Given the efforts $\boldsymbol{x}$ spent by all the agents, the joint protection of agent i is $\sum_{j=1}^n w_{ij} x_j$ with an interconnected effect parameterized by weights $W = \{w_{ij}\}_{i,j \in [n]}$. The probability of being attacked is modeled by $\sigma(-\sum_{j=1}^n w_{ij} x_j + L_i)$, where σ is the sigmoid function and L_i refers to the value of agent i.

The Stackelberg leader is an external insurer who can customize insurance plans to influence agents' protection decisions. The leader can set an insurance plan $\pi = \{I_i, \rho_i\}_{i \in [n]}$ to agent i, where ρ_i is the premium paid by agent i to receive compensation I_i when attacked. Under the insurance plans and the interconnected effect, agents selfishly determine their effort spent on the protection $\boldsymbol{x}$ to maximize their payoff. On the other hand, the leader's objective is the total premium subtracting the compensation paid, while the constraints on the feasible insurance plans are the individual rationality of each customer, i.e., the compensation and premium must incentivize agents to purchase the insurance plan by making the payoff with insurance no worse than the payoff without. These constraints restrict the premium and compensation offered by the insurer.

Experiments and Discussion

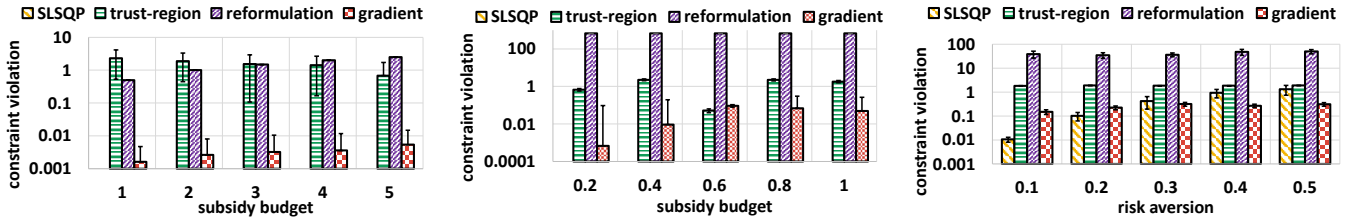
We compare our gradient-based Algorithm 1 (**gradient**) against various baselines in the three settings described above. In each experiment, we execute 30 independent runs (100 runs for SSGs) under different randomly generated instances. We run Algorithm 1 with learning rate $\gamma = 0.01$ for 5,000 gradient steps and update the Lagrange multipliers every $K = 100$ iterations. Our gradient-based method completes in about an hour across all settings—refer to the appendix for more details.

Baselines We compare against several baselines that can solve the stochastic Stackelberg problem with multiple followers with equilibrium-dependent objective and constraints. In particular, given the non-convexity of agents' objective functions, SSGs and cyber insurance games can have multiple, stochastic equilibria. Our first baseline is the leader's **initial** strategy π_0, which is a naive all-zero strategy in all three settings. Blackbox optimization baselines include sequential least squares programming (**SLSQP**) (Kraft et al. 1988) and the **trust-region** method (Conn, Gould, and Toint 2000), where the equilibrium encoded in the optimization problem is treated as a blackbox that needs to be repeatedly queried. **Reformulation**-based algorithm (Basilico et al. 2020; Aussel et al. 2020) is the state-of-the-art method to solve Stackelberg games with multiple followers. This approach reformulates the followers' equilibrium conditions into non-linear complementary constraints as a mathematical program with equilibrium constraints (Luo, Pang, and Ralph 1996), then solves the problem using branch-and-bound and mixed integer non-linear programming (we use a commercial solver, Knitro (Nocedal 2006)). The reformulation-based approach cannot handle arbitrary stochastic equilibria but can handle optimistic or pessimistic equilibria. We implement the optimistic version of the reformulation as our baseline, which could potentially suffer from a performance drop as exemplified in Theorem 1.



(a) Normal-form games (b) Multi-defender SSGs (c) Cyber insurance games

Figure 3: We plot the solution quality of the Stackelberg problems with multiple followers. In all three domains, our gradient-based method achieves significantly higher objective than all other approaches. In NFGs and SSGs, the baselines cannot meaningfully improve upon the default strategy of the leader's initialization due to the high dimensionality of the parameter π; in cyber insurance games, SLSQP and reformulation both make some progress but still mostly with lower utility.



(a) Normal-form games (b) Multi-defender SSGs (c) Cyber insurance games

Figure 4: We plot the average budget constraint violation. Our gradient-based approach maintains low violation across all settings. SLSQP produces no violation in the first two domains because it fails to provide any meaningful improvement against the leader's initialization. Other baselines violate constraints more (often by orders of magnitude) despite less performance improvement.

Solution Quality

In Figure 3(a) and 3(b), we plot the leader's objective (y-axis) versus various budgets for the paid subsidy (x-axis). Figure 3(c), shows the total revenue to the insurer (y-axis) versus the risk aversion of agents (x-axis). Denoting the number of agents by n and the number of actions per agent by m, we have $n = 3, 5, 10$ and $m = 10, 50, 1$ in NFGs, SSGs, and cyber insurance games, respectively.

Our optimization baselines perform poorly in Figure 3(a) and 3(b) due to the high dimensionality of the environment parameter π in NFGs ($\dim(\pi) = nm^n$) and SSGs ($\dim(\pi) = nm$), respectively. In Figure 3(c), the dimensionality of cyber insurance games ($\dim(\pi) = 2n$) is smaller, where we can see that SLSQP and reformulation-based approaches start making some progress, but still less than our gradient-based approach. The main reason that blackbox methods do not work is due to the expensive computation of numerical gradient estimates. On the other hand, reformulation method fails to handle the mixed-integer non-linear programming problem reformulated from followers' best response and the constraints within a day.

Constraint Violation

In Figure 4, we provide the average constraint violation across different settings. Blackbox optimization algorithms either become stuck at the initial point due to the inexact numerical gradient estimate or create large constraint violations due to the complexity of equilibrium-dependent constraints. The reformulation approach also creates large constraint violations due to the difficulty of handling large number of non-convex followers' constraints under high-dimensional leader's strategy. In comparison, our method can handle equilibrium-dependent constraints by using an augmented Lagrangian method with an ability to tighten the budget constraint violation under a tolerance as shown. Although Figure 4 only plots the budget constraint violation, in our algorithm, we enforce that the equilibrium oracle runs until the equilibrium constraint violation is within a small tolerance 10^{-6}, whereas other algorithms sometimes fail to satisfy such equilibrium constraints.

Conclusion

In this paper, we present a gradient-based approach to solve Stackelberg games with multiple followers by differentiating through followers' equilibrium to update the leader's strategy. Our approach generalizes to stochastic gradient descent when the equilibrium reached by followers is stochastically chosen from multiple equilibria. We establish the unbiasedness of the stochastic gradient by the equilibrium flow derived from a partial differential equation. To our knowledge, this work is the first to establish the unbiasedness of gradient computed from stochastic sample of multiple equilibria. Empirically, we implement our gradient-based algorithm on three different examples, where our method outperforms existing optimization and reformulation baselines.

Acknowledgement

This research was supported by MURI Grant Number W911NF-17-1-0370. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

References

Agrawal, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, J. Z. 2019. Differentiable convex optimization layers. In *NeurIPS*.

Amos, B.; and Kolter, J. Z. 2017. OptNet: Differentiable optimization as a layer in neural networks. *ICML*.

Aussel, D.; Brotcorne, L.; Lepaul, S.; and von Niederhäusern, L. 2020. A trilevel model for best response in energy demand-side management. *EJOR*.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2019. Deep equilibrium models. In *NeurIPS*.

Basilico, N.; Coniglio, S.; and Gatti, N. 2017. Methods for finding leader–follower equilibria with multiple followers. *arXiv preprint arXiv:1707.02174*.

Basilico, N.; Coniglio, S.; Gatti, N.; and Marchesi, A. 2020. Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games. *EJCO*.

Bertsekas, D. P. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Blum, A.; Haghtalab, N.; Hajiaghayi, M.; and Seddighin, S. 2019. Computing Stackelberg Equilibria of Large General-Sum Games. In *International Symposium on Algorithmic Game Theory*, 168–182. Springer.

Calvete, H. I.; and Galé, C. 2007. Linear bilevel multi-follower programming with independent followers. *Journal of Global Optimization*, 39(3): 409–417.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research*, 153(1): 235–256.

Coniglio, S.; Gatti, N.; and Marchesi, A. 2020. Computing a pessimistic stackelberg equilibrium with multiple followers: The mixed-pure case. *Algorithmica*, 82(5): 1189–1238.

Conn, A. R.; Gould, N. I.; and Toint, P. L. 2000. *Trust region methods*. SIAM.

Dafermos, S. 1988. Sensitivity analysis in variational inequalities. *Mathematics of Operations Research*, 13(3): 421–434.

Facchinei, F.; Kanzow, C.; and Sagratella, S. 2014. Solving quasi-variational inequalities via their KKT conditions. *Mathematical Programming*, 144(1-2): 369–412.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; Lemieux, A.; et al. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security. In *AAAI*.

Gan, J.; Elkind, E.; Kraus, S.; and Wooldridge, M. 2020. Mechanism Design for Defense Coordination in Security Games. In *AAMAS*.

Gan, J.; Elkind, E.; and Wooldridge, M. 2018. Stackelberg security games with multiple uncoordinated defenders. *AAMAS*.

Hu, M.; and Fukushima, M. 2011. Variational inequality formulation of a class of multi-leader-follower games. *Journal of optimization theory and applications*, 151(3): 455–473.

Jiang, A. X.; Procaccia, A. D.; Qian, Y.; Shah, N.; and Tambe, M. 2013. Defender (mis) coordination in security games. *IJCAI*.

Johnson, B.; Böhme, R.; and Grossklags, J. 2011. Security games with market insurance. In *GameSec*. Springer.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2010. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24.

Kraft, D.; et al. 1988. A software package for sequential quadratic programming.

Krawczyk, J. B.; and Uryasev, S. 2000. Relaxation algorithms to find Nash equilibria with economic applications. *Environmental Modeling & Assessment*, 5(1): 63–73.

Kuhn, H. W.; and Tucker, A. W. 2014. Nonlinear programming. In *Traces and emergence of nonlinear programming*, 247–258. Springer.

Li, J.; Yu, J.; Nie, Y.; and Wang, Z. 2020. End-to-End Learning and Intervention in Games. *NeurIPS*.

Li, Z.; Jia, F.; Mate, A.; Jabbari, S.; Chakraborty, M.; Tambe, M.; and Vorobeychik, Y. 2021. Solving Structured Hierarchical Games Using Differential Backward Induction. *arXiv preprint arXiv:2106.04663*, abs/2106.04663.

Lina, M.; and Jacqueline, M. 1996. Hierarchical systems with weighted reaction set. In *Nonlinear optimization and Applications*, 271–282. Springer.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2018. What game are we playing? end-to-end learning in normal and extensive form games. *IJCAI*.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2019. Large scale learning of agent rationality in two-player zero-sum games. In *AAAI*.

Liu, B. 1998. Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computers & Mathematics with Applications*, 36(7): 79–89.

Luo, Z.-Q.; Pang, J.-S.; and Ralph, D. 1996. *Mathematical programs with equilibrium constraints*. Cambridge University Press.

McKelvey, R. D.; and Palfrey, T. R. 1995. Quantal response equilibria for normal form games. *Games and economic behavior*.

Mertikopoulos, P.; and Zhou, Z. 2019. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1): 465–507.

Mortensen, D. T.; and Pissarides, C. A. 2001. Taxes, subsidies and equilibrium labour market outcomes. *Available at SSRN 287319*.

Naghizadeh, P.; and Liu, M. 2014. Voluntary participation in cyber-insurance markets. In *Workshop on the Economics of Information Security (WEIS)*.

Nakamura, T. 2015. One-leader and multiple-follower Stackelberg games with private information. *Economics Letters*, 127: 27–30.

Nocedal, J. 2006. KNITRO: an integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization*, 35–60. Springer.

Parise, F.; and Ozdaglar, A. 2019. A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. *Games and Economic Behavior*, 114: 47–82.

Pirnay, H.; López-Negrete, R.; and Biegler, L. T. 2012. Optimal sensitivity based on IPOPT. *Mathematical Programming Computation*, 4(4): 307–331.

Rotemberg, M. 2019. Equilibrium effects of firm subsidies. *American Economic Review*, 109(10): 3475–3513.

Roughgarden, T. 2004. Stackelberg scheduling strategies. *SIAM journal on computing*, 33(2): 332–350.

Shi, C.; Zhang, G.; and Lu, J. 2005. The Kth-best approach for linear bilevel multi-follower programming. *Journal of Global Optimization*, 33(4): 563–578.

Sinha, A.; Malo, P.; Frantsev, A.; and Deb, K. 2014. Finding optimal strategies in a multi-period multi-leader–follower Stackelberg game using an evolutionary algorithm. *Computers & Operations Research*, 41: 374–385.

Sinha, A.; Soun, T.; and Deb, K. 2019. Using Karush-Kuhn-Tucker proximity measure for solving bilevel optimization problems. *Swarm and evolutionary computation*, 44: 496–510.

Solis, C. U.; Clempner, J. B.; and Poznyak, A. S. 2016. Modeling multileader–follower noncooperative Stackelberg games. *Cybernetics and Systems*, 47(8): 650–673.

Ui, T. 2016. Bayesian Nash equilibrium and variational inequalities. *Journal of Mathematical Economics*, 63: 139–146.

Uryasev, S.; and Rubinstein, R. Y. 1994. On relaxation algorithms in computation of noncooperative equilibria. *IEEE Transactions on Automatic Control*, 39(6): 1263–1267.

Zhang, H.; Xiao, Y.; Cai, L. X.; Niyato, D.; Song, L.; and Han, Z. 2016. A multi-leader multi-follower Stackelberg game for resource management in LTE unlicensed. *IEEE Transactions on Wireless Communications*, 16(1): 348–361.

Implementation Details

We implement a differentiable PyTorch module to compute a sample of the followers' equilibria. The module takes the leader's strategy as input and outputs a Nash equilibrium computed in the forward pass using the relaxation algorithm. We use a random initialization to run the relaxation algorithm, which can reach to different equilibria depending on different initialization. Given the sampled equilibrium $\boldsymbol{x}^*$ computed in the forward pass, the backward pass is implemented by PyTorch autograd to compute all the second-order derivatives to express Equation 5. The backward pass solves the linear system in Equation 5 analytically to derive $\frac{d\boldsymbol{x}^*}{d\pi}$ as an approximate of the equilibrium flow.

This PyTorch module is used in all three examples in our experiment. The implementation is flexible as we just need to adjust the followers' objectives and constraints, the relaxation algorithm and the gradient computation all directly apply.

Proofs of Theorem 2 and Theorem 3

Theorem 2. *If $v(\boldsymbol{x}^*, \pi)$ is the equilibrium flow of the stochastic equilibrium oracle $\mathcal{O}(\pi)$, we have:*

$$\begin{aligned}&\frac{d}{d\pi}\,\mathbb{E}_{\boldsymbol{x}^*\sim\mathcal{O}(\pi)}\,f(\boldsymbol{x}^*,\pi)\\ =&\,\mathbb{E}_{\boldsymbol{x}^*\sim\mathcal{O}(\pi)}\left[f_\pi(\boldsymbol{x}^*,\pi)+f_{\boldsymbol{x}}(\boldsymbol{x}^*,\pi)\cdot v(\boldsymbol{x}^*,\pi)\right].\end{aligned}\tag{8}$$

Proof. To compute the derivative on the left-hand side, we have to first expand the expectation because the equilibrium distribution is dependent on the environment parameter π:

$$\begin{aligned}&\frac{d}{d\pi}\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f(\boldsymbol{x},\pi)\\ =&\frac{d}{d\pi}\int_{\boldsymbol{x}\in\mathcal{X}}f(\boldsymbol{x},\pi)p(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&\int_{\boldsymbol{x}\in\mathcal{X}}\left(p(\boldsymbol{x},\pi)\frac{\partial}{\partial\pi}f(\boldsymbol{x},\pi)+f(\boldsymbol{x},\pi)\frac{\partial}{\partial\pi}p(\boldsymbol{x},\pi)\right)d\boldsymbol{x}\\ =&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f_\pi(\boldsymbol{x},\pi)+\int_{\boldsymbol{x}\in\mathcal{X}}f(\boldsymbol{x},\pi)\frac{\partial}{\partial\pi}p(\boldsymbol{x},\pi)d\boldsymbol{x}\end{aligned}\tag{11}$$

We further define $\Phi(\boldsymbol{x},\pi)=p(\boldsymbol{x},\pi)v(\boldsymbol{x},\pi)$. By the equilibrium flow definition in Equation 7, we have

$$\frac{\partial}{\partial\pi}p(\boldsymbol{x},\pi)=-\nabla_{\boldsymbol{x}}\cdot\Phi(\boldsymbol{x},\pi)$$

Therefore, the later term in Equation 11 can be computed by integration by parts and Stokes' theorem:

$$\begin{aligned}&\int_{\boldsymbol{x}\in\mathcal{X}}f(\boldsymbol{x},\pi)\frac{\partial}{\partial\pi}p(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&-\int_{x\in\mathcal{X}}f(\boldsymbol{x},\pi)\nabla_{\boldsymbol{x}}\cdot\Phi(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&-\int_{x\in\mathcal{X}}\nabla_{\boldsymbol{x}}\cdot(f(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi))d\boldsymbol{x}\\ &\qquad+\int_{\boldsymbol{x}\in\mathcal{X}}f_{\boldsymbol{x}}(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&-\oint_{\partial\mathcal{X}}f(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi)dS+\int_{\boldsymbol{x}\in\mathcal{X}}f_{\boldsymbol{x}}(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi)d\boldsymbol{x}\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\frac{d}{d\pi}\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f(\boldsymbol{x},\pi)\\ =&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f_\pi(\boldsymbol{x},\pi)+\int_{\boldsymbol{x}\in\mathcal{X}}f(\boldsymbol{x},\pi)\frac{\partial}{\partial\pi}p(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f_\pi(\boldsymbol{x},\pi)-\oint_{\partial\mathcal{X}}f(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi)dS\\ &\qquad+\int_{\boldsymbol{x}\in\mathcal{X}}f_{\boldsymbol{x}}(\boldsymbol{x},\pi)\Phi(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}f_\pi(\boldsymbol{x},\pi)-\oint_{\partial\mathcal{X}}f(\boldsymbol{x},\pi)p(\boldsymbol{x},\pi)v(\boldsymbol{x},\pi)dS\\ &\qquad+\int_{\boldsymbol{x}\in\mathcal{X}}f_{\boldsymbol{x}}(\boldsymbol{x},\pi)p(\boldsymbol{x},\pi)v(\boldsymbol{x},\pi)d\boldsymbol{x}\\ =&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{O}(\pi)}\left[f_\pi(\boldsymbol{x},\pi)+f_{\boldsymbol{x}}(\boldsymbol{x},\pi)v(\boldsymbol{x},\pi)\right]\end{aligned}$$

where the term $\oint_{\partial\mathcal{X}}f(\boldsymbol{x},\pi)p(\boldsymbol{x},\pi)v(\boldsymbol{x},\pi)dS=0$ because $p(\boldsymbol{x},\pi)=0$ at the boundary $\partial\mathcal{X}$. This concludes the proof of Theorem 2. □

Notice that in the proof of Theorem 2, we only use integration by parts and Stokes' theorem, where both of them apply to Riemann integral and Lebesgue integral. Thus, the proof of Theorem 2 also works for any measure zero jumps in the probability density function.

Theorem 3. *Given the leader's strategy π and a sampled equilibrium $\boldsymbol{x}$, if (1) the KKT matrix at $(\boldsymbol{x},\pi)$ is invertible and (2) $\boldsymbol{x}$ is sampled with a fixed probability locally, the solution to Equation 5 is a homogeneous solution to Equation 7 and matches the equilibrium flow $v(\pi,\boldsymbol{x})$ locally.*

Proof. Since the KKT conditions hold for all equilibria, the given π and $\boldsymbol{x}$ must satisfy $KKT(\boldsymbol{x},\pi)=0$. The KKT matrix in Equation 5 is given by $\frac{\partial KKT}{\partial\boldsymbol{x}}$, the Jacobian matrix of the function $KKT(\boldsymbol{x},\pi)$ with respect to $\boldsymbol{x}$. If the KKT matrix is invertible, by implicit function theorem, there exists an open set U containing π such that there exists a unique continuously differentiable function $h:U\to\mathcal{X}$ such that $h(\pi)=\boldsymbol{x}$ and $KKT(h(\pi'),\pi')=0$ for all $\pi'\in U$. Moreover, the analysis in Equation 5 applies, where $\frac{dh(\pi)}{d\pi}=\frac{d\boldsymbol{x}}{d\pi}$ matches the solution of Equation 5.

Lastly, the condition that the equilibrium $\boldsymbol{x}$ is sampled with a fixed probability density c locally implies the corresponding probability density function must satisfy $p(x',\pi')=c\mathbf{1}_{\mathrm{KKT}(x',\pi')=0}=c\mathbf{1}_{\boldsymbol{x}'=h(\pi')}$ for all $\pi'\in U$ in an open set locally[3].

Now we can verify whether $p(\boldsymbol{x}',\pi')$ and $v(\boldsymbol{x}',\pi')=\frac{dh(\pi')}{d\pi}$ (independent of $\boldsymbol{x}'$) satisfy the partial differential equation of equilibrium flow as defined in Definition 3. We

[3]We can choose the smaller subset U such that both the implicit function theorem and the locally fixed probability c both hold.

first compute the left-hand side of Equation 7 by:

$$\frac{\partial}{\partial \pi} p(\boldsymbol{x}', \pi') = \frac{\partial}{\partial \pi} c \mathbf{1}_{\boldsymbol{x}'=h(\pi')} = c\delta_{\boldsymbol{x}'=h(\pi')} \frac{dh(\pi')}{d\pi} \tag{12}$$

where Equation 12 is derived by fixing $\boldsymbol{x}'$, the derivative of a jump function $\mathbf{1}_{\boldsymbol{x}'=h(\pi')}$ is a Dirac delta function located at $\boldsymbol{x}' = h(\pi')$ multiplied by a Jacobian term $\frac{dh(\pi')}{d\pi}$.

We can also compute the right-hand side of Equation 7 by:

$$\begin{aligned}
&\nabla_x \cdot (p(\boldsymbol{x}', \pi') v(\boldsymbol{x}', \pi')) \\
=&v(\boldsymbol{x}', \pi') \frac{\partial}{\partial \boldsymbol{x}} p(\boldsymbol{x}', \pi') + p(\boldsymbol{x}', \pi') \frac{\partial}{\partial \boldsymbol{x}} v(\boldsymbol{x}', \pi') && (13) \\
=&\frac{dh(\pi')}{d\pi} \frac{\partial}{\partial \boldsymbol{x}} c \mathbf{1}_{\boldsymbol{x}'=h(\pi')} \\
=&c\delta_{\boldsymbol{x}'=h(\pi')} \frac{dh(\pi')}{d\pi} && (14)
\end{aligned}$$

where the second term in Equation 13 is 0 because we define $v(\boldsymbol{x}', \pi') = \frac{dh(\pi')}{d\pi}$, which is independent of $\boldsymbol{x}'$. Equation 14 is derived by fixing π', the derivative of a jump function is a Dirac delta function located at $\boldsymbol{x}' = \pi'$.

The above calculation shows that Equation 12 is identical to Equation 14, which implies the left-hand side and the right-hand side of Equation 7 are equal. Therefore, we conclude that the choice of $v(\boldsymbol{x}', \pi') = \frac{d\boldsymbol{x}'}{d\pi} = \frac{dh(\pi')}{d\pi}$ is a homogeneous solution to differential equation in Equation 7 locally in $\pi' \in U$. By the definition of the equilibrium flow, $v(\boldsymbol{x}', \pi') = \frac{d\boldsymbol{x}'}{d\pi}$ is a solution to the equilibrium flow because we can subtract the homogeneous solution and define a new partial differential equation without region U to compute the solution outside of U. □

Limitation of Theorem 2 and Theorem 3

Although Theorem 2 always holds, the main challenge preventing us from directly applying Theorem 2 is that we do not know the equilibrium flow in advance. Given the probability density function of the equilibrium oracle, we can compute the equilibrium flow by solving the partial differential equation in Equation 7. However, the probability density function is generally not given.

Theorem 3 tells us that the derivative computed in Equation 5 is exactly the equilibrium flow defined by the partial differential equation when the sampled equilibrium admits to an invertible KKT matrix and is locally sampled with a fixed probability. That is to say, when these conditions hold, we can treat the equilibrium sampled from a distribution over multiple equilibria as a unique equilibrium to differentiate through as discussed in the section of unique Nash equilibrium. These conditions are also satisfied when the sampled equilibrium is locally stable without any discontinuous jump, generalizing the differentiability of unique Nash equilibrium and globally isolated Nash equilibria to the case with only conditions on the sampled Nash equilibrium.

Dimensionality and Computation Cost

Dimensionality of Control Parameters

We discuss the solution quality attained and computation costs required by different optimization methods. To understand the results, it is useful to compare the role and dimensionality of the environment parameter π in each setting.

- *Normal-form games*: parameter π corresponds to the non-negative subsidies provided to each follower for each entry of its payoff matrix. We have $\dim(\pi) = n \prod_{i=1}^{n} m_i = nm^n$, where for simplicity we set $m_i = m$ for all i.
- *Stackelberg security games*: parameter π refers to the non-negative subsidies provided to each follower at each available target. Because each follower i can only cover targets $T_i \subseteq T$, we have $\dim(\pi) = \sum_{i=1}^{n} |T_i| = nm$, where we set $|T_i| = m$ for all i.
- *Cyber insurance games*: each insurance plan is composed of a premium and a coverage amount. Therefore in total, $\dim(\pi) = 2n$, the smallest out of the three tasks.

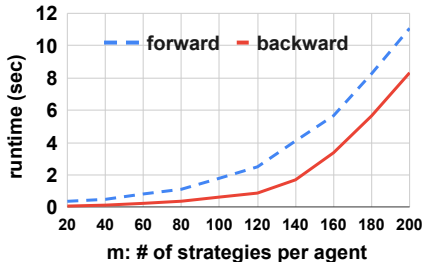
Computation Cost

In Figure 5, we compare the computation cost per iteration of equilibrium-finding oracle (forward) and the gradient oracle (backward). Due to the hardness of the Nash equilibrium-finding problem, no equilibrium oracle is likely to have polynomial-time complexity in the forward pass (computing an equilibrium). We instead focus more on the computation cost of the backward pass (differentiating through an equilibrium).

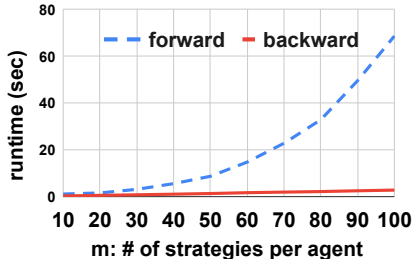
As we can see in Equation 5, the complexity of gradient computation is dominated by inverting the KKT matrix with size $L = O(nm)$ and the dimensionality of environment parameter π since the matrix $\frac{d\boldsymbol{x}^*}{d\pi}$ is of size $L \times \dim(\pi)$. Therefore, the complexity of the backward pass is bounded above by $O(L^\alpha) + O(L^2 \dim(\pi)) = O(n^\alpha m^\alpha) + O(n^2 m^2 \dim(\pi))$ with $\alpha = 2.373$.

- In Figure 5(a), the complexity is given by $O(n^2 m^2 \dim(\pi)) = O(n^3 m^{n+2}) = O(m^5)$ where we set $n = 3$ with varied m, number of actions per follower, shown in the x-axis.
- In Figure 5(b), the complexity is $O(n^2 m^2 \dim(\pi)) = O(m^3)$ with $n = 5$ and varied m, number of actions per follower, shown in the x-axis.
- In Figure 5(c), the complexity is $O(n^2 m^2 \dim(\pi)) = O(n^3)$ with $m = 1$ and varied number of followers n shown in the x-axis. The runtime of the forward pass increases drastically, while the runtime of the backward pass remains polynomial.

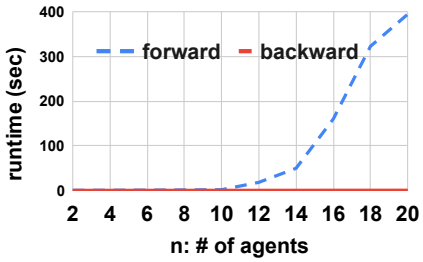
In all three examples, the gradient computation (backward) has polynomial complexity and is faster than the equilibrium finding oracle (forward). Numerical gradient estimation in gradient-free methods requires repeatedly accessing the forward pass, which can be even more expensive than our gradient computation.



(a) Normal-form games with $n = 3$ followers and varied m strategies per follower.



(b) Stackelberg security games with $n = 5$ followers and varied m strategies per follower.



(c) Cyber insurance games with $m = 1$ dimensional investment decision and varied n.

Figure 5: We compare the computation cost of equilibrium computation (forward) and the gradient access (backward) per iteration. Backward pass is cheaper than forward pass in all three domains. Gradient-based method runs a forward pass and a backward pass per iteration, while gradient-free method requires many forward passes to perform one step of local search.

Optimization Reformulation of the Stackelberg Problems with Multiple Followers

In this section, we describe how to reformulate the leader's optimization problem with multiple followers involved into an single-level optimization problem with stationary and complementarity constraints. Notice that this reformulation requires the assumption that all followers break ties in favor of the leader, while our gradient-based method can deal with arbitrary oracle access not limited to any tie-breaking rules.

Normal-Form Games with Risk Penalty

In this example, the followers' objectives are defined by:

$$f_i(\boldsymbol{x}, \pi) = U_i(\boldsymbol{x}) + \pi_i(\boldsymbol{x}) - H(x_i)/\lambda, \qquad (15)$$

where U_i is the given payoff matrix and π_i is the subsidy provided by the leader. H is the Gibbs entropy denoting the risk aversion penalty.

The leader's objective and the constraint are respectively defined by:

$$f(\boldsymbol{x}, \pi) = \sum\nolimits_{i\in[n]} U_i(\boldsymbol{x})$$
$$g(\boldsymbol{x}, \pi) = \left(\sum\nolimits_{i\in[n]} \pi_i(\boldsymbol{x})\right) - B \le 0.$$

Bilevel optimization formulation we can write the followers' best response into the leader's optimization problem:

$$\begin{aligned}
\max_{\boldsymbol{\pi}} \quad & f(\boldsymbol{x}) = \sum\nolimits_{i\in[n]} U_i(\boldsymbol{x}) = U(\boldsymbol{x}) \\
\text{s.t.} \quad & x_i \in [0,1]^{m_i}, \mathbf{1}^\top x_i = 1 && \forall i \in [n] \\
& x_i = \arg\max_{x\in\mathcal{X}_i} f_i(x_i, x_{-i}, \boldsymbol{\pi}) && \forall i \in [n] \\
& \pi(\boldsymbol{x}) \le B
\end{aligned}$$

where f_i is defined in Equation 15. By converting the inner-level optimization problem to its KKT conditions, we can rewrite the optimization problem as:

$$\begin{aligned}
\min_{\boldsymbol{\pi},\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}} \quad & -f(\boldsymbol{x}) = -U(\boldsymbol{x}) \\
\text{s.t.} \quad & x_i, \quad \mathbf{1}^\top x_i = 1 && \forall i \in [n] \\
& \lambda_i, \mu_i \in \mathbb{R}^{m_i}_{\ge 0}, \nu_i \in \mathbb{R} && \forall i \in [n] \\
& \lambda_{i,j} x_{i,j} = 0 && \forall i \in [n], j \in [m_i] \\
& \mu_{i,j}(1 - x_{i,j}) = 0 && \forall i \in [n], j \in [m_i] \\
& -\nabla_{x_i} f_i - \lambda_i + \mu_i + \nu_i \mathbf{1} = 0 && \forall i \in [n] \\
& \pi(\boldsymbol{x}) \le B
\end{aligned}$$

We add dual variables λ_i, μ_i to the inequality constraints $x_{i,j} \ge 0$ and $x_{i,j} \le 1$ respectively. We also add dual variables ν_i to the equality constraints $\mathbf{1}^\top x_i = 1$. We can explicitly write down the gradient:

$$\nabla_{x_i} f_i(x_i, x_{-i}, \boldsymbol{\pi}) = (U_i + \boldsymbol{\pi}_i)(x_{-i}) - \sum\nolimits_j (1 + \log x_{ij})/\lambda \qquad (16)$$

where λ here is a specific constant (different from the Lagrangian multipliers), which is chosen to be 1 in our implementation.

Stackelberg Security Games With Multiple Defenders

The followers' objectives are defined by:

$$f_i(\boldsymbol{x}, \pi) = \sum\nolimits_{t\in T_i} (U_{i,t} + \pi_{i,t})(1 - y_t) p_t, \qquad (17)$$

where $U_{i,t}$ is the loss received by defender i when target t is successfully attacked, and $\pi_{i,t}$ is the corresponding reimbursement provided by the leader to remedy the loss. We define $y_t := 1 - \prod_i (1 - x_{i,t})$ to denote the effective coverage of target t, representing the probability that target t is protected under the overlapping protection patrol plan $\boldsymbol{x}$. Given the effective coverage of all targets, we assume the attacker attacks target t with probability $p_t = e^{-\omega y_t + a_t} / (\sum_{s\in T} e^{-\omega y_s + a_s})$, where $a_t \in \mathbb{R}$ is a known attractiveness value and $\omega \ge 0$ is a scaling constant.

The leader's objective and constraint are respectively defined by:

$$f(\boldsymbol{x}, \pi) = \sum\nolimits_{t \in T} U_t(1 - y_t)p_t$$
$$g(\boldsymbol{x}, \pi) = \Big(\sum\nolimits_{i,t} \pi_{i,t}(1 - y_t)p_t\Big) - B \leq 0,$$

where $U_t < 0$ is the penalty for the leader when target t is attacked without any coverage.

Bilevel optimization formulation Similarly, we can also write down the bilevel optimization formulation of the Stackelberg security games with multiple defenders as:

$$\begin{aligned}
\max_{\boldsymbol{\pi}} \quad & f(\boldsymbol{x}) = \sum\nolimits_{t \in T} U_t(1 - y_t)p_t \\
\text{s.t.} \quad & x_{i,t} \in [0, 1] && \forall i \in [n], t \in T_i \\
& y_t, p_t \in \mathbb{R} && \forall t \in T \\
& \sum\nolimits_{t \in T_i} x_{i,t} = b_i && \forall i \in [n] \\
& y_t = 1 - \prod\nolimits_{i:t \in T_i} (1 - x_{i,t}) && \forall t \in T \\
& p_t = \frac{e^{-\omega y_t + a_t}}{\sum_{s \in T} e^{-\omega y_s + a_s}} && \forall t \in T \\
& x_i = \arg\max_{x \in \mathcal{X}_i} f_i(x_i, x_{-i}, \boldsymbol{\pi}) && \forall i \in [n] \\
& \sum_{i,t} \big(\pi^u_{i,t}(1 - y_t)p_t + \pi^c_{i,t} y_t p_t\big) \leq B
\end{aligned}$$

where p_t is the probability that attacker will attack target t under protect scheme $\boldsymbol{x}$ and the resulting $\boldsymbol{y}$. The function f_i is defined in by:

$$f_i(\boldsymbol{x}, \pi) = \sum\nolimits_{t \in T_i} (U_{i,t} + \pi_{i,t})(1 - y_t)p_t. \quad (18)$$

This bilevel optimization problem can be reformulated into a single level optimization problem if we assume all the individual followers break ties (equilibria) in favor of the leader, which is given by:

$$\begin{aligned}
\max_{\boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}} \quad & \sum\nolimits_{t \in T} U_t(1 - y_t)p_t \\
\text{s.t.} \quad & x_{i,t} \in [0, 1] && \forall i \in [n], t \in T_i \\
& y_t, p_t \in \mathbb{R} && \forall t \in T \\
& \sum\nolimits_{t \in T_i} x_{i,t} = b_i && \forall i \in [n] \\
& y_t = 1 - \prod\nolimits_{i:t \in T_i} (1 - x_{i,t}) && \forall t \in T \\
& p_t = \frac{e^{-\omega y_t + a_t}}{\sum_{s \in T} e^{-\omega y_s + a_s}} && \forall t \in T \\
& \lambda_{i,t}, \mu_{i,t} \in \mathbb{R}_{\geq 0}, \nu_i \in \mathbb{R}_{\geq 0} && \forall i \in [n], t \in T_i \\
& \lambda_{i,t} x_{i,t} = 0 && \forall i \in [n], t \in T_i \\
& \mu_{i,t}(1 - x_{i,t}) = 0 && \forall i \in [n], t \in T_i \\
& -\nabla_{x_i} f_i - \lambda_i + \mu_i + \nu_i \mathbf{1} = 0 && \forall i \in [n] \\
& \sum_{i,t} \big(\pi^u_{i,t}(1 - y_t)p_t + \pi^c_{i,t} y_t p_t\big) \leq B
\end{aligned}$$

Similarly, we add dual variables $\lambda_{i,t}, \mu_{i,t}, \nu_i$ to constraints $x_{i,t} \geq 0$, $x_{i,t} \leq 1$, and $\sum_{t \in T_i} x_{i,t} = b_i$.

Cyber Insurance Games

The followers' objectives are defined by:

$$f_i(\boldsymbol{x}, \pi) = -c_i x_i - \rho_i - (L_i - I_i)q_i - \gamma |L_i - I_i| \sqrt{q_i(1 - q_i)}, \quad (19)$$

where c_i is the unit cost of the protection x_i and L_i is the loss when the computer is attacked. The insurance plan offered to agent i is denoted by (ρ_i, I_i), where ρ_i is the fixed premium paid to enroll in the insurance plan and I_i is the compensation received when the computer is attacked.

We assume the computer is attacked with a probability q_i, where $q_i = \sigma(-\sum_{j=1}^n w_{ij} x_j + v L_i)$ with σ being sigmoid function, a matrix $W = \{w_{ij} > 0\}_{i,j \in [n]}$ to represent the interconnectedness between agents, $v \geq 0$ to reflect the attacker's preference over high-value targets, and lastly it depends on the loss L_i incurred by agent i when attacked. This attack probability is a smooth non-convex function, which makes the reformulation approach hard and the non-convexity can lead to multiple equilibria reached by the followers.

The last term in Equation 19 is the risk penalty to agent i. This term is the standard deviation of the loss received by agent i. We assume the agent is risk averse and thus penalized by a constant time of the standard deviation.

On the other hand, the leader's objective is defined by:

$$f(\boldsymbol{x}, \pi) = \sum\nolimits_{i=1}^{n} -I_i q_i + \rho_i$$

where the leader's objective is simply the total revenue received by the insurer, which includes the premium collected from all agents and the compensation paid to all agents.

The constraints are the individual rationality of each agent, where the customized insurance plan needs to incentivize the agent to purchase the insurance plan. In other words, the compensation I_i and premium ρ_i must incentivize agents to purchase the insurance plan by making the payoff with insurance no worse than the payoff without.

$$g_i(\boldsymbol{x}, \pi) = \Big(-c_i x_i - L_i q_i - \gamma L_i \sqrt{q_i(1 - q_i)}\Big) - f_i(\boldsymbol{x}, \pi) \leq 0.$$

Bilevel optimization reformulation The bilevel optimization formulation for the cyber insurance domain with an external insurer is given by:

$$\begin{aligned}
\max_{\boldsymbol{\pi}} \quad & f(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n} -I_i q_i + \rho_i \\
\text{s.t.} \quad & x_i \in [0, \infty) && \forall i \in [n] \\
& q_i = \sigma\Big(-\sum\nolimits_{j=1}^{n} w_{ij} x_j + v L_i\Big) && \forall i \in [n] \\
& x_i = \arg\max_{x'_i \in \mathcal{X}_i} f_i(x'_i, x_{-i}, \pi) && \forall i \in [n] \\
& -c_i x_i - L_i q_i - \gamma L_i \sqrt{q_i(1 - q_i)} \leq f_i(\boldsymbol{x}, \pi) && \forall i \in [n]
\end{aligned}$$

where $f_i(\boldsymbol{x}, \pi) = -c_i x_i - \rho_i - (L_i - I_i)q_i - \gamma \|L_i - I_i\| \sqrt{q_i(1 - q_i)}$.

Reformulating this bilevel problem into a single level optimization problem, we have:

$$
\begin{aligned}
\max_{\boldsymbol{\pi},\boldsymbol{x},\lambda} \quad & f(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n} -I_i q_i + \rho_i \\
\text{s.t.} \quad & x_i \in [0,\infty), \lambda_i \in [0,\infty) && \forall i \in [n] \\
& q_i = \sigma\left(-\sum\nolimits_{j=1}^{n} w_{ij} x_j + v L_i\right) && \forall i \in [n] \\
& x_i \lambda_i = 0 && \forall i \in [n] \\
& -c_i x_i - L_i q_i - \gamma L_i \sqrt{q_i(1-q_i)} \le f_i(\boldsymbol{x}, \pi) && \forall i \in [n] \\
& -\nabla_{x_i} f_i - \lambda_i = 0 && \forall i \in [n]
\end{aligned}
$$

with dual variables λ_i for the $x_i \geq 0$ constraint.

Experimental Setup

For reproducibility, we set the random seeds to be from 1 to 30 for NSGs and cyber insurance games, and from 1 to 100 for SSGs.

Normal-Form Games

In NFGs, we randomly generate the payoff matrix $U_i \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$ of follower i with each entry of the payoff matrix randomly drawn from a uniform distribution $U(0, 10)$. We assume there are $n = 3$ followers. Each follower has three pure strategies to use $m_i = m = 3$ for all i. The risk aversion penalty constant is set to be $\lambda = 1$.

Stackelberg Security Games

In SSGs, we randomly generate the penalty $U_{i,t} < 0$ of each defender i associated to each target $t \in T_i \subset T$ from a uniform distribution $U_{i,t} \sim U(-10, 0)$. The leader's penalty $U_t < 0$ is also generated from the same uniform distribution $U_t \sim U(-10, 0)$. We assume there are $n = 5$ followers in total. There are $|T| = 100$ targets and each follower is able to protect $|T_i| = m = 50$ targets randomly sampled from all targets. Each follower can spend at most $b_i = 10$ effort on the available targets. The attractiveness values a_t used to denote the attacker's preference is randomly generated from a normal distribution $a_t \in \mathcal{N}(0, 1)$ with 0 mean and standard deviation 1. The scaling constant is set to be $\omega = 5$.

Cyber Insurance Games

In cyber insurance games, for each follower i, we generate the unit protection cost c_i from a uniform distribution $c_i \sim U(5, 10)$, and the incurred loss L_i from a uniform distribution $L_i \sim U(50, 100)$. We assume there are in total $n = 10$ followers. Each follower can only determine their own investment and thus $m = 1$. The entry of the correlation matrix $W \in \mathbb{R}^{n \times n}$ is generated from uniform distributions $W_{i,j} \sim U(0, 1)$ if $i \neq j$, and $W_{i,j} \sim U(1, 2)$ if $i = j$ to reflect the higher dependency on the self investments. We choose the risk aversion constant γ to be $\gamma = 0.01$.

Computing Infrastructure

All experiments except VI experiments were run on a computing cluster, where each node is configured with 2 Intel Xeon Cascade Lake CPUs, 184 GB of RAM, and 70 GB of local scratch space. VI experiments require a Knitro license and were run on a machine with i9-7940X CPU @ 3.10GHz with 14 cores and 128 GB of RAM. Within each experiment, we did not implement parallelization, so each experiment was purely run on a single CPU core.