The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Robust Stackelberg Equilibria in Extensive-Form Games and Extension to Limited Lookahead

Christian Kroer, Gabriele Farina, Tuomas Sandholm
Computer Science Department, Carnegie Mellon University
{ckroer,gfarina,sandholm}@cs.cmu.edu

Abstract

Stackelberg equilibria have become increasingly important as a solution concept in computational game theory, largely inspired by practical problems such as security settings. In practice, however, there is typically uncertainty regarding the model about the opponent. This paper is, to our knowledge, the first to investigate Stackelberg equilibria under uncertainty in extensive-form games, one of the broadest classes of game. We introduce robust Stackelberg equilibria, where the uncertainty is about the opponent's payoffs, as well as ones where the opponent has limited lookahead and the uncertainty is about the opponent's node evaluation function. We develop a new mixed-integer program for the deterministic limited-lookahead setting. We then extend the program to the robust setting for Stackelberg equilibrium under unlimited and under limited lookahead by the opponent. We show that for the specific case of interval uncertainty about the opponent's payoffs (or about the opponent's node evaluations in the case of limited lookahead), robust Stackelberg equilibria can be computed with a mixed-integer program that is of the same asymptotic size as that for the deterministic setting.

In a Stackelberg equilibrium, a *leader* commits to a strategy first, and then a *follower* chooses a strategy for herself. The model was first introduced in the context of competition between firms where the leader picks a quantity to supply, and then the follower picks a quantity to supply (von Stackelberg 1934). Stackelberg equilibria have become important as a solution concept in computational game theory, largely inspired by practical problems such as security settings, where the leader is a defender who picks a mixed (i.e., potentially randomized) strategy first, and then the follower who is the attacker decides where to attack, if at all.

Most work on Stackelberg equilibria has focused on *normal-form* (aka. matrix-form) games. Conitzer and Sandholm (2006) studied the problem of computing an optimal strategy to commit to in normal-form games. That line of work has been extended to many security-game applications. In practice, there is typically uncertainty about the opponent's payoffs. In normal-form games this has been studied as *Bayesian Stackelberg games* where the players have private information about their own payoffs, and there is common knowledge of the prior distribution over the payoffs (Tambe 2011; Paruchuri et al. 2008). As an alternative, the *robust* (distribution-free) approach has been suggested for security games: bounds are assumed on the follower's payoffs (Kiekintveld, Islam, and Kreinovich 2013; Nguyen et al. 2014).

Extensive-form games (EFGs)—i.e., tree-form games—are a very general game representation language. EFGs are exponentially more compact and also more expressive than normal-form games. Letchford and Conitzer (2010) study how to compute an optimal strategy to commit to in EFGs and prove hardness results under several assumptions about the game structure. Bošanskỳ et al. (2015) provide further results specifically for perfect-information EFGs. Bosansky and Cermak (2015) develop a *mixed-integer program (MIP)* for computing a Stackelberg strategy, and Cermak et al. (2016) develop an iterative approach based on upper-bounding solutions from extensive-form *correlated* Stackelberg equilibria.

To our knowledge, we are the first to consider uncertainty about the opponent in Stackelberg strategies for EFGs. This is important because EFGs are a powerful representation language and because in practice there is typically uncertainty about the opponent. We take a robust approach to modeling this uncertainty. We introduce robust Stackelberg equilibria for EFGs, where the uncertainty is about the opponent's payoffs, as well as ones where the opponent has limited lookahead and the uncertainty is about the opponent's node evaluation function.

We develop a new MIP for the deterministic limited-lookahead setting. We then extend the MIP to the robust setting for Stackelberg equilibrium under unlimited and under limited lookahead by the opponent. We show that for the specific case of interval uncertainty about the opponent's payoffs (or about the opponent's node evaluations in the case of limited lookahead), robust Stackelberg equilibria can be computed with a MIP that is of the same asymptotic size as that for the deterministic setting.

Our results for robust Stackelberg equilibria in EFGs are relevant to security-game settings with sequential interactions, where EFG models can more compactly represent certain games, as compared to a normal-form representation (Bosansky and Cermak 2015). Robust models are important in security games, where opponent models often have uncertainty, both in standard security games (Kiek-

intveld, Tambe, and Marecki 2010; Kiekintveld, Islam, and Kreinovich 2013; Nguyen et al. 2014), and *green security games* (Nguyen et al. 2015).

Our limited-lookahead results are useful for settings where it is not always desirable to model adversaries as fully rational, but as having limited lookahead capability. This includes settings such as biological games, where the goal is to steer an evolutionary process or an adaptation process which typically acts myopically without lookahead (Sandholm 2015; Kroer and Sandholm 2016b) and security games where opponents are often assumed to be myopic (which can be especially well motivated when the number of adversaries is large (Yin et al. 2012) or in the case of *opportunistic criminals* (Zhang et al. 2016; Rosenfeld and Kraus 2017)). Our model of limited lookahead is an extension of that of Kroer and Sandholm (2015) to a robust setting. Kroer and Sandholm (2015) gave a MIP for computing an optimal strategy to commit to in the deterministic setting. We show an alternative MIP for computing such a strategy to commit to, which we then extend to the robust setting.

Finally, the question of robust variants of optimization problems has been studied extensively in the optimization literature (Ben-Tal and Nemirovski 2002; Ben-Tal, El Ghaoui, and Nemirovski 2009; Bertsimas, Brown, and Caramanis 2011). In that literature, the assumption is that we are given some *nominal* mathematical program, and then the robust variant requires that each constraint in the nominal program holds with respect to every instantiation of a set of uncertainty parameters. This makes the setting substantially different from our setting, where there is no nominal program: the best response of the follower does not need to be a best response for every uncertainty instantiation (this would be the equivalent to robust optimization, and often infeasible), but rather the best response is chosen after the uncertainty parameters are chosen.

Extensive-Form Games

Extensive-form games (EFGs) can be thought of as a game tree, where each node in the tree corresponds to some history of actions taken by all players. Each node belongs to some player, and the actions available to the player at a given node are represented by the branches. Uncertainty is modeled by having a special player, *Nature*, that moves with some predefined fixed probability distribution over actions at each node belonging to Nature. EFGs model imperfect information by having groups of nodes in *information sets*, where an information set is a group of nodes all belonging to the same player such that the player cannot distinguish among them. Finally we assume *perfect recall*, which requires that no player forgets information they knew earlier in the game.

Definition 1. *A* leader-follower two-player extensive-form game with imperfect information and perfect recall Γ *is a tuple* $(H, Z, A, P, f_c, \mathcal{I}_l, \mathcal{I}_f, u_l, u_f)$ *composed of:*

- H*: a finite set of possible sequences (or histories) of actions, such that the empty sequence* $\varnothing \in H$*, and every prefix* z *of* h *in* H *is also in* H*.*
- $Z \subseteq H$*: the set of terminal histories, i.e., those sequences that are not a proper prefix of any sequence.*
- A*: a function mapping* $h \in H \setminus Z$ *to the set of available actions at non-terminal history* h*.*
- P*: the player function, mapping each non-terminal history* $h \in H \setminus Z$ *to* $\{l, f, c\}$*, representing the player whose turn it is to move after history* h*. If* $P(h) = c$*, the player is Chance.*
- $\mathcal{C}$*: a function assigning to each* $h \in H$ *the probability of reaching* h *due to nature (i.e. assuming that both players play to reach* h*).*
- $\mathcal{I}_i$*, for* $i \in \{l, f\}$*: partition of* $\{h \in H : P(h) = i\}$ *with the property that* $A(h) = A(h')$ *for each* h, h' *in the same set of the partition. For notational convenience, we will write* $A(I)$ *to mean* $A(h)$ *for any of the* $h \in I$*, where* $I \in \mathcal{I}_i$*.* $\mathcal{I}_i$ *is the information partition of player* i*, while the sets in* $\mathcal{I}_i$ *are called the information sets of player* i*.*
- u_i*: utility function mapping* $z \in Z$ *to the utility gained by player* i *when the terminal history is reached.*

We further assume that all players have perfect recall.

We will use the more relaxed term *extensive-form game*, or EFG, to mean a two-player extensive-form game with imperfect information and perfect recall.

In this paper we will investigate settings where there is uncertainty about the follower's utility function u_f. Specifically, the follower's utility can be any function from some given *uncertainty set* U_f consisting of functions that map from the set of leaf nodes to $\mathbb{R}$. We leave the exact structure of U_f undefined for now; in our algorithmic section we show that the case where each leaf has independent interval uncertainty can be solved using a MIP.

A strategy for a player i is usually represented in *behavioral form*, which consists of probability distributions over actions at each information set in $\mathcal{I}_i$. In this paper we will focus on an alternative, but strategically equivalent, representation of the set of strategies, called the *sequence form* (Romanovskii 1962; Koller, Megiddo, and von Stengel 1996; von Stengel 1996). In the sequence form, actions are instead represented by *sequences*. A sequence σ_i, is an ordered list of actions taken by player i on the path to some history h. In perfect-recall games, all nodes in an information set $I \in \mathcal{I}_i$ correspond to the same sequence for player i. We let $\sigma(I)$ denote this sequence. Given a sequence σ_i and an action a that Player i can take immediately after σ_i, we let $\sigma_i a$ denote the resulting new sequence. The set of all sequences for player i is denoted by Σ_i. Instead of directly choosing the probability to put on an action, in the sequence form the probability of playing the entire sequence is chosen; this is called the *realization probability* and is denoted by $r_i(\sigma_i)$. A choice of realization probabilities for every sequence belonging to Player i is called a *realization plan* and is denoted by $r_i : \Sigma_i \to [0, 1]^{|\Sigma_i|}$. This representation relies on perfect recall: for any information set $I \in \mathcal{I}_i$ we have that each action $a \in A(I)$ is uniquely represented by a single sequence $\sigma_i = \sigma(I)a$, since $\sigma(I)$ corresponds to exactly one sequence. This gives us a simple way to convert any strategy in sequence form to a behavioral strategy: the probability of playing action $a \in A(I)$ at information set I is simply $\frac{r_i(\sigma(I)a)}{r_i(\sigma(I))}$. For a sequence $\sigma = \sigma' a$, we let the information

set such that $a \in A(I), \sigma(I) = \sigma'$ be denoted by $\inf(\sigma)$.

It will be convenient to have function expressing expected values for a given pair of sequences. Given two sequences σ_l and σ_f, we let

$$g_l(\sigma_l, \sigma_f) = \sum_{h \in Z; \sigma_f(h) = \sigma_f; \sigma_l(h) = \sigma_l} \mathcal{C}(h) u_l(h),$$

$$g_f^{u_f}(\sigma_l, \sigma_f) = \sum_{h \in Z; \sigma_f(h) = \sigma_f; \sigma_l(h) = \sigma_l} \mathcal{C}(h) u_f(h)$$

be the expected utilities, for the leader and follower respectively, over leaf nodes that are reached with σ_f, and σ_l as the corresponding last player sequences. The function for the follower $g_f^{u_f}$ depends on the choice of utility function u_f, whereas we always know the utility function for the leader.[1] Given two realization plans r_l, r_f and a utility function u_i, we overload notation slightly and let the expected value for Player i induced by the realization plans be denoted by

$$u_i(r_l, r_f) = \sum_{\sigma_l \in \Sigma_l, \sigma_f \in \Sigma_f} r_l(\sigma_l) r_f(\sigma_f) g_i(\sigma_l, \sigma_f).$$

Stackelberg Setting

We will focus on settings where the leader first commits to a strategy that the follower observes. The follower then plays a best response to the leader strategy. A *strong Stackelberg equilibrium* (SSE) is a pair of strategies r_l, r_f such that r_f is a best response to r_l and r_l is a solution to the optimization problem of maximizing $u(r_l, r_f)$ over r_l and r_f, subject to the constraint that r_f is a best response to r_l. This definition implies the common assumption that the follower breaks ties in favor of Player l (Tambe 2011; Conitzer and Sandholm 2006; Paruchuri et al. 2008). A *weak Stackelberg equilibrium* assumes minimization over the set of optimal best responses.

Limited-Lookahead Model

We will also consider a limited-lookahead variant of EFGs. There has been a significant amount of work on limited lookahead in perfect-information games (such as chess and checkers) in the AI community. Modeling limited lookahead in imperfect-information games (that have information sets) is more intricate. A model for that was presented recently (Kroer and Sandholm 2015), and we use that model. In that model, the follower can only look ahead k steps. He uses a *node-evaluation function* $\tilde{u} : H \to \mathbb{R}$ that associates a heuristic utility with any node in the game tree. At any information set $I \in \mathcal{I}_f$, the follower has a set of nodes $\tilde{H}_I \subset H$ called the *lookahead frontier*. When choosing his action at information set I, the follower chooses an action that maximizes the expected value of $\tilde{u}$, assuming that they choose actions so as to maximize $\tilde{u}$ at any follower information sets reached before $\tilde{H}_I$. We let $g_I(\sigma_l, \sigma_f)$ be the expected value over lookahead-frontier nodes according to the node-evaluation function (analogous to g_i for the setting without limited lookahead). We assume that for any information set $I' \in \mathcal{I}_f$ that comes after I, all the nodes of I' are entirely contained in the set of nodes that precede $\tilde{H}_I$, or entirely disjoint with the set of preceding nodes (this is in order to avoid any information sets belonging to the follower being only partially contained in the hypothetical decision making under I). We let the set of information sets that come after I such that their nodes are all preceding $\tilde{H}_I$ be denoted by $\mathcal{I}_I$. We $\Sigma_f^I \subseteq \Sigma_f$ denote the set of all sequences beneath a given information set I that are within the lookahead frontier.

In the prior paper on limited lookahead in imperfect-information games (Kroer and Sandholm 2015) it was assumed that the leader knows the follower's node evaluation function exactly. That seems quite unrealistic. Therefore, we will extend the work to the case where the leader has uncertainty about the follower's node evaluation function.

Best Responses and how to Compute Them

Our solution concept will depend on the notion of a *best response* for the follower. For a given leader strategy r_l and utility function $u_f \in U_f$, the set of best responses is

$$BR(r_l, u_f) = \{r_f : u_f(r_l, r_f) = \max_{r_f'} u_f(r_l, r_f')\}.$$

Given a strategy r_l for the leader and a utility function u_f, the value of each information set can be computed with the following feasibility program (this holds outside of a leader-follower setting as well):

$$v_{\inf_f(\sigma_f)} = s_{\sigma_f} + \sum_{\substack{I' \in \mathcal{I}_f \\ \sigma_f(I') = \sigma_f}} v_{I'} + \sum_{\sigma_l \in \Sigma} r_l(\sigma_l) g_f^{u_f}(\sigma_l, \sigma_f) \qquad \forall I \in \mathcal{I}_f, \sigma_f = \sigma_f(I) \quad (1)$$

$$0 \leq s_{\sigma_f} \leq M(1 - b_f(\sigma_f)) \qquad \forall \sigma_f \in \Sigma_f \quad (2)$$

$$\textstyle\sum_{a \in A(I)} b_f(\sigma a) = 1 \qquad \forall I \in \mathcal{I}_f, \sigma_f = \sigma_f(I) \quad (3)$$

$$b_f(\sigma_f) \in \{0, 1\} \qquad \sigma_f \in \Sigma_f \quad (4)$$

The variables v_I represent the value of a given information set I, $b_f(\sigma_f)$ represents whether σ_f is a best response at its respective information set, and s_{σ_f} represents how much less utility the follower gets by following the sequence σ_f rather than the optimal action at $\inf(\sigma_f)$. It is easy to show via induction that the feasibility MIP given in equations (1-4) computes a best response to r_l and the variables v_I represent the values of information sets I when best-responding to r_l: For the base case of an information set with no future information sets belonging to the follower, disregarding s_{σ_f}, the RHS of (1) clearly represents the value of choosing σ_f at the information set. Now, since all s_{σ_f} are nonnegative and (1) is an equality, it follows that v_I upper bounds the value of each individual sequence at I. But since $s_{\sigma_f} = 0$ for some σ_f, it must be an equality for said σ_f. Thus v_I upper bounds the value of all sequences at I, but is also equal to the value of some sequence, and therefore it represents the value when best responding. Applying the inductive hypothesis to any information set I that has future information

[1] In Stackelberg equilibrium, the follower does not have to be concerned about the leader's utility function because the leader commits to his strategy and declares his strategy to the follower.

sets belonging to the follower reduces the expression for v_I to one that is equivalent to the base case.

Extension to Uncertainty about the Opponent

We now extend the EFG model to incorporate uncertainty about the follower's utility function. We will take a robustness approach, where we care about the worst-case instantiation of the uncertainty set U_f. For limited-lookahead EFGs we will analogously consider uncertainty over the node-evaluation function.

Due to the uncertainty (represented by the uncertainty set U_f), defining a Stackelberg equilibrium is not straightforward. We take the perspective that a robust Stackelberg solution is a strategy for the leader that maximizes the leader utility in the worst-case instantiation of U_f:

Definition 2. *A* robust strong Stackelberg solution *(RSSS) is a realization plan r_l such that*

$$r_l \in \arg\max_{r'_l \in R_l} \inf_{u_f \in U_f} \max_{r'_f \in BR(r_l, u_f)} u_l(r_l, r'_f).$$

The robustness is represented by the minimization over U_f. Intuitively, if the actual instantiation of u_f does not take on the minimizer over U_f, the leader can only receive better utility, so we are computing the maximin utility against the robustness. Typically one is interested in finding an RSSS strategy for the leader, but we nonetheless define the entire equilibrium concept as well:

Definition 3. *A* robust strong Stackelberg equilibrium *(RSSE) is a realization plan r_l and a (potentially uncountably large) set of realization plans $\{r_f^{u_f} : \forall u_f \in U_f\}$ such that r_l is an RSSS and $r_f^{u_f} \in BR(r_l, u_f)$ for all $u_f \in U_f$.*

Whether an RSSE is even practical to represent is highly dependent on the structure of the specific game and uncertainty sets at hand, as it would frequently need to be represented parametrically. On the other hand, once we have r_l, the best response for a specific u_f can easily be computed. One method for doing this is to first compute the follower value u^* under u_f when best responding to r_l (e.g., via a single tree traversal), and then solving the *linear program (LP)* that consists of maximizing the leader's utility over the set of follower strategies that achieve u^* (this can be done by adding a single constraint to the sequence-form best-response LP given by von Stengel (1996)).

One might consider applying the robustness after the follower chooses her strategy (in a sense, swapping the inner max and min). In this case, we cannot represent this as a minimization on the inside since the set of best responses is defined with respect to the choice of u_f. Arguably the most natural way to apply robustness after the best response of the follower would be to ask for a pair of strategies r_l, r_f such that r_f is a best response no matter the instantiation of u_f. This definition of robustness would allow us to apply standard robust optimization techniques to any Stackelberg MIP. However, this definition has several drawbacks. First, if we are applying a robust model, we are often interested in maximizing our worst-case utility. By applying robustness after choosing r_f, we would not be doing that, but instead would be maximizing utility subject to the constraint that we want to be sure what the follower response is. Second, a robust Stackelberg equilibrium defined that way would not necessarily exist: if there is overlap between the range of possible utilities associated with a pair of actions at some information set, there would be no way to guarantee that a single action will always be a best response.

MIP for Full-Certainty Setting

We now give a MIP for computing a Stackelberg equilibrium in a game where the follower has limited lookahead.

$$\max_{p,r,v,s} \sum_{z \in Z} p(z) u_l(z) \mathcal{C}(z) \tag{5}$$

$$v_I = s_{\sigma_f} + \sum_{I' \in \mathcal{I}_f : \sigma_f(I') = \sigma_f} v_{I,I'} + \sum_{\sigma_l \in \Sigma_l} r_l(\sigma_l) g_I(\sigma_l, \sigma_f) \quad \forall \sigma_f \in \Sigma_f, I = \inf_f(\sigma_f) \tag{6}$$

$$v_{I,\inf(\sigma_f)} = s^I_{\sigma_f} + \sum_{I' \in \mathcal{I}_f : \sigma_f(I') = \sigma_f} v_{I,I'} + \sum_{\sigma_l \in \Sigma_l} r_l(\sigma_l) g_I(\sigma_l, \sigma_f) \quad \forall I \in \mathcal{I}, \sigma_f \in \Sigma^I_f \tag{7}$$

$$r_i(\emptyset) = 1 \qquad \forall i \in \{l, f\} \tag{8}$$

$$r_i(\sigma_i) = \textstyle\sum_{a \in A(I_i)} r_i(\sigma_i a) \qquad \forall i \in \{l, f\}, I_i \in \mathcal{I}_i \tag{9}$$

$$0 \le s_{\sigma_f} \le (1 - r_f(\sigma_f))\, M \qquad \forall \sigma_f \in \Sigma_f \tag{10}$$

$$0 \le s^I_{\sigma_f} \le \left(1 - r^I_f(\sigma_f)\right) M \qquad \forall I \in \mathcal{I}_f, \sigma^I_f \in \Sigma^I_f \tag{11}$$

$$r_f(\sigma_f) \in \{0, 1\} \qquad \forall \sigma_f \in \Sigma_f \tag{12}$$

$$r^I_f(\sigma_f) \in \{0, 1\} \qquad \forall I \in \mathcal{I}_f \sigma_f \in \Sigma^I_f \tag{13}$$

$$0 \le p(z) \le r_i(\sigma_i(z)) \qquad \forall i \in \{l, f\}, z \in Z \tag{14}$$

$$1 = \textstyle\sum_{z \in Z} p(z) \mathcal{C}(z) \tag{15}$$

$$0 \le r_l(\sigma_l) \le 1 \qquad \forall \sigma_l \in \Sigma_l \tag{16}$$

This MIP is an extension of the MIP given by Bosansky and Cermak (2015) to the limited-lookahead setting of Kroer and Sandholm (2015). Eq. (5) is the expected leader value over leaf nodes. Equations (6) to (13) set up best-response constraints for each follower information set, as well as for each pair of information sets I, I' such that $I' \in \mathcal{I}_I$ (these constraints are completely analogous to (1)-(4) except that the constraints involving $v_{I,I'}$ must be set up for each I in order to represent best responses when applying the lookahead evaluation function at I). Equations (14) and (15) ensure that the probabilities over leaves are correct. Finally (8), (9), and (16) ensure that r_l is a valid leader strategy.

MIP with Uncertainty about Follower Payoff

We now move to the computation of RSSS for the setting with uncertainty about follower payoff but no limited lookahead. We will consider a particular class of uncertainty functions: interval uncertainty on each leaf payoff. More concretely, the uncertainty set will be

$$U_f = \{u_f : u_f(h) \in [L(h), U(h)], \forall h \in Z\},$$

where $L(h), U(h)$ are given upper and lower bounds on the interval that the payoff for leaf node h must be chosen from.

One issue that now arises is that we may not be able to make a single action optimal: if the maximum-to-minimum utility intervals for two sequences are guaranteed to overlap we cannot make either sequence the optimal choice for the follower player. Instead, we allow choosing both sequences, and we then assume that the leader player receives the minimum over the two. Intuitively, this can be thought of as a zero-sum game played within the space of actions made optimal for the follower player (a similar technique was used in Kroer and Sandholm (2015)). More generally, we may have $k > 1$ actions at a given information set that can all be made optimal under various instantiations of the utility function. We now introduce a set-valued function that, under some given strategy for the follower r_f, returns the set of actions at a given follower information set that can be made optimal under some instantiation of the utility function, given the tie-breaking rule,

$$A_I(r_f) = \{a \in A_I : \nexists\, a' \in A_I,\ v_I^L(a') \geq v_I^U(a)\}.$$

For any $a \in A_I(r_f)$, the minimization over the uncertainty can choose an instantiation making a the only best-response action at I. Conversely, for $a \notin A_I(r_f)$, even if the utility function is chosen to maximize the value of action a, there exists some other action a' whose worst-case instantiation is at least as good; if a leads to better leader utility than a' then the minimization over utility functions will not allow them to be tied, and if a leads to worse utility than a', then even if a utility function causing a tie is chosen, the best-response tie-breaking in favor of the leader means that a' will be chosen. Thus, a' (or some other action) is always chosen over a.[2]

The function $A_I(r_f)$ is illustrated in Figure 1. The general intuition can be seen from the figure: the dotted line denotes the split between potentially optimal actions (black bars) and actions that cannot be made optimal through any utility-function choice (opaque bars). Note that the dotted line is touched by interval end-points from both sets: this means that the two actions could be tied, but the lower-value would never be chosen, since it is either worse for the leader, in which case the tie-breaking does not choose it even in case of a tie, or if it is better then the minimization over the intervals will break the tie and make it inoptimal.

The intuition behind our robust MIP consists of three components: 1) the best-response feasibility MIP described in (1)-(4), instantiated independently for both the set of maximal and minimal valuation functions, 2) a set of constraints for computing the set $A_I(r_f)$ for a given r_f via best-response values for the maximal and minimal utility functions, and 3) a minimization similar to the dual best-response LP from the standard sequence-form LP (von Stengel 1996).

In the robust MIP given below, g_f^U, g_f^L are the functions giving the expected value over leaf nodes consistent with a pair of sequences, when every node has its payoff set to the maximal (g_f^U) and minimal (g_f^U) payoff, respectively.

[2]Here we rely on the assumption that every action has a strict inequality $v_I^U(a) > v_I^L(a)$. Without this assumption our MIP still works, but the math becomes more cumbersome.

Figure 1: A set of action value-uncertainty intervals.

$$\min_{r,v,s,y} y_0 \tag{17}$$

$$y_{\inf_f(\sigma_f)} \geq \sum_{\substack{I' \in \mathcal{I}_f \\ \sigma_f(I') = \sigma_f}} y_{I'} - \sum_{\sigma_l \in \Sigma_l} g_l(\sigma_l, \sigma_f) r_l(\sigma_l) - M(1 - r_f(\sigma_f)) \quad \forall \sigma_f \in \Sigma_f \tag{18}$$

$$v^q_{\inf_f(\sigma_f)} = s^q_{\sigma_f} + \sum_{\substack{I' \in \mathcal{I}_f \\ \sigma_f(I') = \sigma_f}} v^q_{I'} + \sum_{\sigma_l \in \Sigma} r_l(\sigma_l) g_f^q(\sigma_l, \sigma_f) \quad \forall \sigma_f \in \Sigma_f, q \in \{U, L\} \tag{19}$$

$$0 \leq s^q_{\sigma_f} \leq M(1 - b^q_f(\sigma_f)) \quad \forall \sigma_f \in \Sigma_f, q \in \{U, L\} \tag{20}$$

$$\textstyle\sum_{a \in A(I)} b^q_f(\sigma_f(I)a) = 1 \quad \forall q \in \{U, L\}, I \in \mathcal{I}_f \tag{21}$$

$$b^q_f(\sigma_f) \in \{0, 1\} \quad \forall \sigma_f \in \Sigma_f, q \in \{U, L\} \tag{22}$$

$$v_I^U - s^U_{\sigma_f} \geq v_I^L - M(1 - r_f(\sigma_f)) \quad \forall \sigma_f \in \Sigma_f \tag{23}$$

$$v_I^U - s^U_{\sigma_f} \leq v_I^L + M r_f(\sigma_f) \quad \forall \sigma_f \in \Sigma_f \tag{24}$$

$$r_i(\emptyset) = 1 \quad \forall i \in \{l, f\} \tag{25}$$

$$r_l(\sigma) = \sum_{a \in A(I)} r_l(\sigma a) \quad \forall I \in \mathcal{I}_l, \sigma = \sigma_l(I) \tag{26}$$

$$r_f(\sigma) \leq \sum_{a \in A(I)} r_f(\sigma a) \quad \forall I \in \mathcal{I}_f, \sigma = \sigma_f(I) \tag{27}$$

$$r_f(\sigma_f) \in \{0, 1\} \quad \forall \sigma_f \in \Sigma_f \tag{28}$$

$$0 \leq r_l(\sigma_l) \leq 1 \quad \forall \sigma_l \in \Sigma_l \tag{29}$$

Equations (17) and (18) implement the minimization over the set of potentially optimal actions $A_I(r_l)$ at a given information set I. Equations (19) to (22) ensure that v_I^U, v_I^L represent the correct value of each information set under the maximal and minimal utility function. Equations (23) and (24) ensure that actions in or not in $A_I(r_l)$ can potentially be made optimal (23) or cannot be made optimal (24). Equations (25) to (29) ensure that r_l is a valid sequence-form leader strategy and that one more pure strategies are active for the follower. We prove that this MIP computes a RSSS. Due to space constraints the result is shown in the appendix.

MIP for Limited-Lookahead Interval Uncertainty

We also present an extension of the full-certainty MIP for limited lookahead to a setting with uncertainty about the limited-lookahead node-evaluation function. That MIP joins the ideas from both the full-certainty MIP with limited lookahead ((5)-(16)) and the robust MIP ((17)-(29)) and is thus the most comprehensive, but it combines the novel ideas from the former two MIPs in a fairly straightforward way. Due to limited space we present the MIP in the appendix.

Experiments

Using our MIPs presented in the previous section we investigated the scalability and qualitative properties of RSSS solutions. We experimented with three kinds of EFG: Kuhn poker (Kuhn) (Kuhn 1950), a 2-card poker variant (2-card), and a parameterized security-inspired search game (Search). The search game is similar to games considered by Bosansky et al. (2014) and Bosansky and Cermak (2015)).

Kuhn consists of a three-card deck: king, queen, and jack. Each player first has to put a payment of 1 into the pot. Each player is then dealt one of the three cards, and the third is put aside unseen. A single round of betting then occurs (with betting parameter $p = 1$, explained below).

In 2-card, the deck consists of two kings and two jacks. Each player first has to put a payment of 1 into the pot. A private card is dealt to each, followed by a betting round (with betting parameter $p = 2$), then a public card is dealt, followed by another betting round (with $p = 4$).

In both games, each round of betting goes as follows:

- Player 1 can check or bet p.
  - If Player 1 checks Player 2 can check or raise p.
    - If Player 2 checks the betting round ends.
    - If Player 2 raises Player 1 can fold or call.
      - If Player 1 folds Player 2 takes the pot.
      - If Player 1 calls the betting round ends.
  - If Player 1 raises Player 2 can fold or call.
    - If Player 2 folds Player 1 takes the pot.
    - If Player 2 calls the betting round ends.

If no player has folded, a showdown occurs. In Kuhn poker, the player with the higher card wins in a showdown. In 2-card, showdowns have two possible outcomes: one player has a pair, or both players have the same private card. For the former, the player with the pair wins the pot. For the latter the pot is split.

Kuhn poker has 55 nodes in the game tree and 13 sequences per player. The 2-card game tree has 199 nodes, and 57 sequences per player.

The search game is played on the graph shown in Figure 2. It is a simultaneous-move game (which can be modeled as a turn-taking EFG with appropriately chosen information sets). The leader controls two patrols that can each move within their respective shaded areas (labeled P1 and P2), and at each time step the controller chooses a move for both patrols. The follower is always at a single node on the graph, initially the leftmost node labeled S and can move freely to any adjacent node (except at patrolled nodes, the follower cannot move from a patrolled node to another patrolled node). The follower can also choose to wait in place for a time step in order to clean up their traces. If a patrol visits a node that was previously visited by the follower, and the follower did not wait to clean up their traces, they can see that the follower was there. If the follower reaches any of the rightmost nodes they received the respective payoff at the node (5, 10, or 3, respectively). If the follower and any patrol are on the same node at any time step, the follower is captured, which leads to a payoff of 0 for the follower and a payoff of 1 for the leader. Finally, the game times out after k simultaneous moves, in which case the leader receives payoff 0 and the follower receives $-\infty$ (because we are interested in games where the follower attempts to reach an end node). We consider games with k being 5 and 6. We will denote these by Search-5 and Search-6. Search-5 (Search-6) has 87,927 (194,105) nodes and 11,830 and 69 (68,951 and 78) leader and follower sequences.

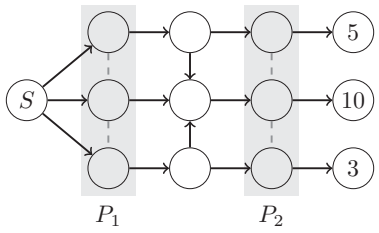

Figure 2: The graph on which the search game is played.

	Kuhn	2-card	Search-5	Search-6
B&C	0	1	13	190
R-0	0	26	1	40
R-0.01	0	548	24	683
R-0.05	0	616	30	910
R-0.1	0	209	36	955
R-0.5	0	365	64	1648
R-1	0	42	41	395

Table 1: Runtime experiments for the MIP by Bosansky and Cermak (2015) (B& C) and our robust Stackelberg MIP for increasing uniform uncertainty intervals (R-c where c is the interval radius). All runtimes are in seconds.

All experiments were conducted using Gurobi 7.5.1 to solve MIPs, on a cluster with 8 Intel Xeon E5607 2.2Ghz cores and 47 GB RAM per experiment.

In the first set of experiments we investigate the impact on runtime caused by uncertainty intervals in each of the four games, without considering limited lookahead. We compare the MIP by Bosansky and Cermak (2015) (B&C) for the full-certainty setting to our robust MIP ((17)-(29)) with an uncertainty interval of diameter d at each node in the game, for 6 different values of d. The results are shown in Table 1. Interestingly, our robust MIP with interval 0 is significantly faster than the B&C MIP for the Search games. (We do not specialize our robust MIP to the full-certainty setting but instead let Gurobi presolve away most redundant variables and constraints. One could easily specialize it and potentially make it even faster.) Once we add uncertainty, the MIP gets harder to solve, with the runtime increasing for larger uncertainty intervals—except for the largest uncertainty interval where the problem starts to get easier again.

In the second set of experiments, we investigate the cost of computing an RSSS against a follower utility function that is different from the one actually employed by the follower. These experiments were conducted on the Search-5 game. On Search-6 it would take prohibitively long to conduct all the experiments, and the experiments would not be interesting on Kuhn and 2-card because they are zero-sum games

	EV	≤ 1	≤ 2	≤ 3
0	0.842	0.474	0.474	0.474
0.1	0.834	0.479	0.479	0.479
0.5	0.800	0.500	0.500	0.500
1	0.758	0.616	0.526	0.526
2	0.688	0.688	0.417	0.417
4	0.667	0.667	0.667	0.667
6	0.667	0.667	0.667	0.667
40	0.500	0.500	0.500	0.500

Table 2: Leader utility when maximizing utility against an incorrect utility function. Each row corresponds to a different size of uncertainty interval used for computing the leader strategy (interval size is given in the leftmost column). The columns are ordered in increasing amounts of incorrectness allowed in the follower utility function.

(the leader will end up getting the value of the game as long as the correct utility function is contained in the uncertainty intervals). The setup is as follows. We use our robust MIP to compute a leader strategy for the original payoffs in Search-5. We instantiate the MIP with several different uncertainty-interval widths (given in the leftmost column in Table 2). For each leader strategy, we then conduct a grid search over triplets of numbers in $\{\pm 0.1, \pm 0.5, \pm 1, \pm 2, \pm 3\}^3$, where the three numbers correspond to a change in utility being added to each of the three rightmost payoff nodes in Figure 2. For each payoff change, we compute the follower's best response (breaking ties in favor of the leader) to the leader strategy under the new game and the resulting leader utility. The second column in Table 2 (EV) denotes the value that the leader is expected to get if he were solving the correct game. The following three columns, labeled $\leq 1, \leq 2, \leq 3$, show the worst utility achieved by the leader when the grid search is restricted to payoff changes of at most 1, 2, and 3, respectively. For example, in the case ≤ 1 we only do the grid search over $\{\pm 0.1, \pm 0.5, \pm 1\}^3$ The experiment shows that when uncertainty is not taken into account, all amounts of perturbation leads to a large decrease in leader utility. Conversely, taking uncertainty into account leads to much better utility in almost every case.

In the third set of experiments, we investigate the cost to the leader from having to take uncertainty into account against a limited-lookahead follower. We perform this experiment on Kuhn and 2-card, both zero-sum games, which allows us to apply the same node-evaluation scheme as in Kroer and Sandholm (2015). In order to construct the limited-lookahead evaluation function, we first compute a Nash equilibrium of the game. We then recursively define the value of each node to be the weighted sum over the values of nodes beneath it, where the weights are the probabilities of each action in the Nash equilibrium, and then add Gaussian noise to the computed value (we do not add any noise to leaf nodes). Since the value of a node is based on the noisy value of nodes beneath it, the farther away from leaf nodes a node is, the noisier the estimate of the node's value (from Nash equilibrium) is. We then use our robust limited-lookahead MIP to solve the limited-lookahead game resulting from having the follower apply this node-evaluation function. We consider lookahead depths of 1 and 2. The results for 2-card are shown in Table 3 and the results for Kuhn are shown in Table 4. The different rows in the tables correspond to varying standard deviations in the Gaussian noise, and columns correspond to increasing sizes of uncertainty intervals. For all games, lookahead depths, and noise levels, we see that the amount that the leader can exploit the follower goes down as uncertainty intervals get larger. However, we also see that for most noise amounts, some amount of robustness can be added without losing substantial leader utility. Coupled with our results from the second set of experiments, which showed that uncertainty intervals are necessary if there is mis-specification in the model, this suggests that uncertainty intervals can lead to substantially more robust outcomes, potentially at a small cost to optimality even if the initial model turns out to be correct.

Lookahead depth: 1

Noise σ	0.01	0.05	0.1	0.5	1	2
0.1	1.35	1.33	1.33	1.15	0.25	0.00
0.5	1.42	1.41	1.33	1.33	0.61	0.00
1	1.50	1.50	1.50	1.33	1.33	0.34
2	1.50	1.50	1.50	1.44	1.33	1.33

Lookahead depth: 2

Noise σ	0.01	0.05	0.1	0.5	1	2
0.1	0.67	0.43	0.05	0.00	0.00	0.00
0.5	0.69	0.69	0.68	0.05	0.00	0.00
1	0.73	0.72	0.71	0.48	0.08	0.00
2	0.80	0.79	0.78	0.71	0.59	0.19

Table 3: Limited-lookahead with depth 1 and 2 in 2-card.

Lookahead depth: 1

Noise σ	0.01	0.05	0.1	0.5	1	2
0.1	0.33	0.33	0.33	0.25	-0.06	-0.06
0.5	0.33	0.33	0.33	0.33	-0.06	-0.06
1	0.33	0.33	0.33	0.33	0.33	0.16
2	0.87	0.87	0.87	0.86	0.84	0.22

Lookahead depth: 2

Noise σ	0.01	0.05	0.1	0.5	1	2
0.1	-0.03	-0.05	-0.06	-0.06	-0.06	-0.06
0.5	0.29	0.28	0.28	0.16	-0.06	-0.06
1	0.41	0.41	0.40	0.30	0.22	0.16
2	0.87	0.87	0.87	0.86	0.84	0.22

Table 4: Limited-lookahead with depth 1 and 2 in Kuhn.

Discussion

While we showed that our technique scales to medium-size games, in practice we would often like to scale to even larger games. The iterative LP-based approach of Cermak et al. (2016) could potentially be extended to the ro-

bust setting. Likewise, abstraction methods have dramatically increased the scalability of Nash equilibrium finding in EFGs (e.g., (Gilpin and Sandholm 2007; Lanctot et al. 2012; Kroer and Sandholm 2014; 2016a; Brown, Ganzfried, and Sandholm 2015)) and could potentially be adapted to the robust Stackelberg setting as well. This could be done while giving guarantees on follower behavior by only abstracting the strategy space of the leader.

Acknowledgements

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1617590, and CCF-1733556, and the ARO under award W911NF-17-1-0082. Christian Kroer is also sponsored by a Facebook Fellowship.

References

Ben-Tal, A., and Nemirovski, A. 2002. Robust optimization–methodology and applications. *Mathematical Programming* 92(3).

Ben-Tal, A.; El Ghaoui, L.; and Nemirovski, A. 2009. *Robust optimization*.

Bertsimas, D.; Brown, D. B.; and Caramanis, C. 2011. Theory and applications of robust optimization. *SIAM review* 53(3).

Bosansky, B., and Cermak, J. 2015. Sequence-form algorithm for computing Stackelberg equilibria in extensive-form games. In *AAAI*.

Bosansky, B.; Kiekintveld, C.; Lisy, V.; and Pechoucek, M. 2014. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research* 829–866.

Bošanskỳ, B.; Brânzei, S.; Hansen, K. A.; Miltersen, P. B.; and Sørensen, T. B. 2015. Computation of Stackelberg equilibria of finite sequential games. In *WINE*.

Brown, N.; Ganzfried, S.; and Sandholm, T. 2015. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit Texas Hold'em agent. In *AAMAS*.

Cermak, J.; Bosansky, B.; Durkota, K.; Lisy, V.; and Kiekintveld, C. 2016. Using correlated strategies for computing Stackelberg equilibria in extensive-form games. In *AAAI*.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *EC*.

Gilpin, A., and Sandholm, T. 2007. Lossless abstraction of imperfect information games. *Journal of the ACM* 54(5).

Kiekintveld, C.; Islam, T.; and Kreinovich, V. 2013. Security games with interval uncertainty. In *AAMAS*.

Kiekintveld, C.; Tambe, M.; and Marecki, J. 2010. Robust Bayesian methods for Stackelberg security games (extended abstract). In *AAMAS*.

Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior* 14(2).

Kroer, C., and Sandholm, T. 2014. Extensive-form game abstraction with bounds. In *EC*.

Kroer, C., and Sandholm, T. 2015. Limited lookahead in imperfect-information games. In *IJCAI*.

Kroer, C., and Sandholm, T. 2016a. Imperfect-recall abstractions with bounds in games. In *EC*.

Kroer, C., and Sandholm, T. 2016b. Sequential planning for steering immune system adaptation. In *IJCAI*.

Kuhn, H. W. 1950. A simplified two-person poker. In *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*.

Lanctot, M.; Gibson, R.; Burch, N.; Zinkevich, M.; and Bowling, M. 2012. No-regret learning in extensive-form games with imperfect recall. In *ICML*.

Letchford, J., and Conitzer, V. 2010. Computing optimal strategies to commit to in extensive-form games. In *EC*.

Nguyen, T. H.; Yadav, A.; An, B.; Tambe, M.; and Boutilier, C. 2014. Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In *AAAI*.

Nguyen, T. H.; Delle Fave, F. M.; Kar, D.; Lakshminarayanan, A. S.; Yadav, A.; Tambe, M.; Agmon, N.; Plumptre, A. J.; Driciru, M.; Wanyama, F.; et al. 2015. Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *GameSec*.

Paruchuri, P.; Pearce, J. P.; Marecki, J.; Tambe, M.; Ordonez, F.; and Kraus, S. 2008. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *AAMAS*.

Romanovskii, I. 1962. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics* 3.

Rosenfeld, A., and Kraus, S. 2017. When security games hit traffic: Optimal traffic enforcement under one sided uncertainty. In *IJCAI*.

Sandholm, T. 2015. Steering evolution strategically: Computational game theory and opponent exploitation for treatment planning, drug design, and synthetic biology. In *AAAI*.

Tambe, M. 2011. Security and game theory: algorithms, deployed systems, lessons learned.

von Stackelberg, H. 1934. *Marktform und Gleichgewicht*. Springer, Vienna.

von Stengel, B. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14(2):220–246.

Yin, Z.; Jiang, A.; Tambe, M.; Kietkintveld, C.; Leyton-Brown, K.; Sandholm, T.; and Sullivan, J. 2012. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*.

Zhang, C.; Gholami, S.; Kar, D.; Sinha, A.; Jain, M.; Goyal, R.; and Tambe, M. 2016. Keeping pace with criminals: An extended study of designing patrol allocation against adaptive opportunistic criminals. *Games* 7(3):15.