Solving Partially Observable Stochastic Games with Public Observations

Karel Horák, Branislav Bošanský
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague
{horak,bosansky}@agents.fel.cvut.cz

Abstract

In many real-world problems, there is a dynamic interaction between competitive agents. Partially observable stochastic games (POSGs) are among the most general formal models that capture such dynamic scenarios. The model captures stochastic events, partial information of players about the environment, and the scenario does not have a fixed horizon. Solving POSGs in the most general setting is intractable. Therefore, the research has been focused on subclasses of POSGs that have a value of the game and admit designing (approximate) optimal algorithms. We propose such a subclass for two-player zero-sum games with discounted-sum objective function—POSGs with *public observations* (PO-POSGs)—where each player is able to reconstruct beliefs of the other player over the unobserved states. Our results include: (1) theoretical analysis of PO-POSGs and their value functions showing convexity (concavity) in beliefs of maximizing (minimizing) player, (2) a novel algorithm for approximating the value of the game, and (3) a practical demonstration of scalability of our algorithm. Experimental results show that our algorithm can closely approximate the value of non-trivial games with hundreds of states.

Introduction

Game theory describes the optimal behavior of rational agents and is recently widely applied to solving security problems. Game-theoretic strategies are used to protect critical infrastructures (Pita et al. 2008; Kiekintveld et al. 2009; Shieh et al. 2012), secure computer networks (Vanek et al. 2012; Nguyen, Wellman, and Singh 2017; Durkota et al. 2017) or wildlife (Fang, Stone, and Tambe 2015; Fang et al. 2016). In many real-world situations, there is a dynamic strategic interaction between the players, and the players do not have perfect information about the environment. Moreover, a pre-defined horizon (number of moves in the scenario) is only rarely given in practice and thus these games belong to the class of partially observable stochastic games (POSGs). Examples include patrolling games (Basilico, Gatti, and Amigoni 2009; Vorobeychik et al. 2014; Basilico, Nittis, and Gatti 2016; Brazdil, Kucera, and Rehak 2018), where a defender protects a set of targets against an attacker, pursuit-evasion (Chung, Hollinger, and Isler 2011), or search games, where a defender is trying to find and capture an attacker.

We focus on two-player zero-sum POSGs, and even with this restriction it is intractable to compute optimal strategies in the most general case. Since the players do not perfectly observe the environment, each player has a belief over possible states of the environment. However, the reward the player receives for choosing some action(s) also depends on the action of the other player who decides based on their belief. Therefore, player 1 has to consider also the belief of player 2 and belief that player 2 has about player 1, and so on. This reasoning is called *nested beliefs* (e.g., in (MacDermed 2013)) and it causes a doubly-exponential number of histories to consider for each agent.

However, real-world security scenarios require partial observability without a strictly defined horizon. Therefore, one can restrict to subclasses of POSGs where games *have a value* (i.e., the value of the game exists) and (approximate) optimal algorithms can be designed. Examples of such works are stochastic games in which both the players' actions and observations are public (Ghosh, McDonald, and Sinha 2004), games in which the support of private/public observations does not depend on states and actions (Cole and Kocherlakota 2001), or games where only one player has imperfect information (also called One-Sided) (Chatterjee and Doyen 2014; Basu and Stettner 2015; Horak, Bosansky, and Pechoucek 2017). The practical motivation for such subclasses is to compute robust strategies for the defender assuming the worst case scenario where the attacker has additional information (Vorobeychik et al. 2014; Horak and Bosansky 2016).

In this paper, we propose a new subclass of POSGs in which we avoid the problem of nested beliefs, called *POSGs with public observations* (PO-POSGs), that generalizes previous subclasses. In this model, each player is able to exactly reconstruct the belief of the opponent. The key characteristics are: (1) the state space is factored – each player observes his private state, but the state of the other player is not observed; (2) each observation that modifies belief about the state of the other player is public (both players are aware of this observation); (3) the true state of the player is observed privately by that player. We restrict to two-players zero-sum games with discounted future rewards and give the following contributions: (1) We

show that games in this class have a value; (2) We show that the value function of PO-POSGs is convex in the belief of the maximizing player and concave in the belief of the minimizing player; (3) We introduce a novel algorithm based on Heuristic Value Iteration Search (HSVI) for One-Sided POSGs (Horak, Bosansky, and Pechoucek 2017; Smith and Simmons 2004) and show that this algorithm converges to the (approximate) optimal values.

We demonstrate our algorithm on two different domains – a patrolling game, where the attacker has imprecise information about the position of the defender (Basilico et al. 2009), and a lasertag game based on a single-player variant (Pineau, Gordon, and Thrun 2003). The results show that, for the first time, there is a practical domain-independent algorithm able to closely approximate optimal values of non-trivial infinite-horizon POSGs with hundreds of states where both players have partial information about the environment.

Related Work

The notion of public actions and observations is common in dynamic games. For finite horizon games, the concept of public states and publicly observed actions creates separated subgames that allow designing limited-lookahead algorithms for imperfect information games (Moravcik et al. 2017; Brown, Sandholm, and Amos 2018).

In games with an infinite horizon, the problem with nested beliefs prevents one from designing an (approximate) optimal algorithm for fully general settings. Nested beliefs can be tackled directly with histories – one of few such approaches is a bottom-up dynamic programming for constructing relevant finite-horizon policy trees for individual players while pruning-out dominated strategies (Hansen, Bernstein, and Zilberstein 2004; Kumar and Zilberstein 2009). However, due to the explicit dependence on the histories, the scalability in the horizon is very limited.

A more common approach is to focus on a subclass of POSGs. In (Ghosh, McDonald, and Sinha 2004), zero-sum POSGs with public actions and observations are considered. The authors show that the value of the game exists and present an algorithm that exploits the transformation of such a model into a game with complete information. In our approach, we assume only public observations (i.e., actions are private to the players). Moreover, we factor the state space according to the players (i.e., each player has his own state that is perfectly observable to this player, and the state of the opponent is unknown). Similar factorization of the state space is used also in (Cole and Kocherlakota 2001), however, in this work the authors assume that the support of observations cannot change due to states or actions of the players. We remove this assumption and actions and observations can be generated in states arbitrarily. Alternatively, some works assume that only one player has partial information (Chatterjee and Doyen 2014; Basu and Stettner 2015; Horak, Bosansky, and Pechoucek 2017). Again, we remove this assumption and allow both players to have partial information about the states of the other player. While our algorithm is based on the algorithm for the one-sided case, we provide significant generalizations of the previous work, especially in the representation of value function, definition and algorithms for computing value-backup operator.

Finally, (MacDermed 2013) gives a transformation of POSGs to Markov Games of Incomplete Information (MaGIIs) as a more efficient representation if observations have Markov property. While the examples of games that we consider satisfy this property, the author demonstrates the benefits of this representation for the common-payoff case of Dec-POMDPs only. We solve zero-sum games, which is a more complex problem and since there is no apparent way to exploit MaGIIs, we use a more common formalism.

POSGs with Public Observations

Definition 1. A partially observable stochastic game with public observations (PO-POSG) is a two-player zero-sum game (played by players $i \in \{1, 2\}$) represented by a tuple $\langle S_i, A_i, O_i, Z_i, T_i, R, b_i^{(0)}, \gamma \rangle$, where

- S_i is a finite set of (private) states of player i
- A_i is a finite set of actions available to player i
- O_i is a finite set of observations for player i
- $Z_i(o_i | s_{-i} a_{-i})$ is the probability to generate observation o_i for player i, given that his opponent[1] $-i$ played an action a_{-i} in state s_{-i}
- $T_i(s_i' | s_i a_i o_i o_{-i})$ is the probability to transition from s_i to s_{-i} when player i played a_i and observations o_i and o_{-i} have been generated
- $R(s_1 s_2 a_1 a_2)$ is the reward of player 1 when actions (a_1, a_2) have been jointly played in the joint state (s_1, s_2)
- $b_i^{(0)} \in \Delta(S_{-i})$ is the initial belief of player i over states S_{-i} of his opponent
- $\gamma \in [0, 1)$ is the discount factor.

A play in a PO-POSG proceeds as follows. First, the initial joint state $(s_1^{(1)}, s_2^{(1)})$ is drawn with probability $b_2^{(0)}(s_1^{(1)}) \cdot b_1^{(0)}(s_2^{(1)})$. Then, in each round t, players observe their current private state (player i observes $s_i^{(t)}$, but not $s_{-i}^{(t)}$ of his opponent). Based on this information (and history), each player i chooses an action $a_i^{(t)} \in A_i$ independently of the decision of his opponent $-i$. As a consequence of this choice, player 1 receives reward $r^{(t)} = R(s_1^{(t)} s_2^{(t)} a_1^{(t)} a_2^{(t)})$ and player 2 receives negated reward $-R(s_1^{(t)} s_2^{(t)} a_1^{(t)} a_2^{(t)})$. Furthermore, an observation $o_i^{(t)}$ for each player is generated and made publicly known to both players with probability $Z_i(o_i^{(t)} | s_{-i}^{(t)} a_{-i}^{(t)})$ and a new private state $s_i^{(t+1)}$ of each player is drawn from $T_i(\cdot | s_i^{(t)} a_i^{(t)} o_i^{(t)} o_{-i}^{(t)})$. We consider discounted setting and the utility of player 1 is thus $\sum_{t=1}^{\infty} \gamma^{t-1} r^{(t)}$ (and negative value for the opponent as the game is zero-sum).

Definition 2. The history of player i up to time T is a sequence $\{s_i^{(t)} a_i^{(t)} o_i^{(t)} o_{-i}^{(t)}\}_{t=1}^{T} s_i^{T+1}$.

Definition 3. The (history-dependent) strategy of player i is a mapping $\sigma_i : (S_i A_i O_i O_{-i})^* S_i \to \Delta(A_i)$ from histories of player i to randomized decisions.

[1] As it is commonly used, $-i$ denotes opponent of player i.

Observe that in PO-POSGs, the player i updates his belief solely on the information about the public observations (o_i, o_{-i}) and the knowledge of the strategy used by the adversary for the *current stage* only—we denote such one-stage strategy by π_{-i} as opposed to the full strategy σ_{-i}. Assuming that the adversary $-i$ chooses an action a_{-i} in a state s_{-i} with probability $\pi_{-i}(a_{-i}|s_{-i})$ in the current stage of the game (given the information available to him) and that observations (o_i, o_{-i}) have been generated, player i can update his belief $b_i \in \Delta(S_{-i})$ to a belief $\tau_{\pi_{-i}}(b_i o_i o_{-i})$ where the updated probability of being in a state s'_{-i} is

$$\tau_{\pi_{-i}}(b_i o_i o_{-i})(s'_{-i}) = \frac{1}{\Pr_{\pi_{-i}}[o_i]} \sum_{s_{-i}, a_{-i}} b_i(s_{-i}) \cdot \tag{1}$$

$$\cdot\, \pi_{-i}(a_{-i}|s_{-i}) \cdot Z(o_i|s_{-i}o_{-i}) \cdot T(s'_{-i}|s_{-i}a_{-i}o_{-i}o_i)\ .$$

Since both the strategy π_{-i} and the public observations (o_i, o_{-i}) are known to player $-i$ as well, she can reconstruct $\tau_{\pi_{-i}}(b_i o_i o_{-i})$, and the belief update is essentially public.

Value of PO-POSGs

We now establish the value function V^* to capture the utility of playing optimal strategies in a PO-POSG (i.e., the value of the game) based on the beliefs the players have.

Definition 4. The optimal *value function* of a PO-POSG is a function $V^* : \Delta(S_2) \times \Delta(S_1) \to \mathbb{R}$ mapping each possible initial belief (b_1, b_2) of the game to the expected utility of player 1 in the equilibrium (i.e., the value of the game).

Since any finite-horizon approximation of a PO-POSGs has a value (von Neumann 1928) and discounted-sum utilities are considered, the value of a PO-POSG is well defined.

Theorem 1. *The value of the game exists in PO-POSGs.*

Proof (sketch). Denote v_T the value of a finite approximation with horizon $T \in \mathbb{N}$. The approximation considers all rewards from the first T steps. The equilibrial strategies in v_T can thus only be inferior in the full, infinite-horizon game, when rewards after T steps are considered. Hence

$$v_T + \textstyle\sum_{t=T+1}^{\infty} \gamma^{t-1} \min R(\cdot) \le V^*[b_1^{(0)}, b_2^{(0)}] \le \\ \le v_T + \textstyle\sum_{t=T+1}^{\infty} \gamma^{t-1} \max R(\cdot)\ . \tag{2}$$

As $T \to \infty$, the bounds converge to $V^*[b_1^{(0)}, b_2^{(0)}]$. □

Contrary to previous works, the optimal value function V^* is neither convex nor concave. We show, however, that due to the factorization of the state space, V^* is convex in the belief b_1 of the maximizing player 1 and concave in the belief b_2 of the minimizing player 2.

Lemma 1. *Let σ_i be a strategy of player i, and b_{-i} be the belief of the adversary. Then the expected utility $V^{\sigma_i|b_{-i}} : \Delta(S_{-i}) \to \mathbb{R}$ of playing σ_i against the best-responding opponent $-i$ parametrized by the belief of player i is linear and $(U - L)$-Lipschitz continuous.*

Proof (sketch). Player $-i$ knows σ_i as well as his true state s_{-i}, and his only uncertainty is about the state s_i (the probability of which is $b_{-i}(s_i)$). It is thus possible to focus on the best response for each state s_{-i} separately. Let us denote the expected utility of playing the best response against σ_i starting from s_{-i} (when $s_i \sim b_{-i}$) by $\xi(s_{-i})$. Since the strategy σ_i is fixed (and thus does not depend on b_i), the expected utility of playing σ_i against the best response of the adversary is the expectation over the values $\xi(s_{-i})$, $V^{\sigma_i|b_{-i}}(b_i) = \sum_{s_{-i}} b_i(s_{-i}) \cdot \xi(s_{-i})$, and thus the value $V^{\sigma_i|b_{-i}}$ is linear in b_i. Moreover, observe that

$$L = \frac{\min R(\cdot)}{1-\gamma} \le V^{\sigma_i|b_{-i}}(b_i) \le \frac{\max R(\cdot)}{1-\gamma} = U \tag{3}$$

which makes V^{σ_i} be $(U - L)$-Lipschitz continuous. □

Theorem 2. *The value function V^* is convex in b_1 and concave in b_2. Moreover, it is $(U - L)\sqrt{2}$-Lipschitz continuous.*

Proof. For a fixed b_2, player 1 chooses a strategy that maximizes the utility, hence

$$V^*[b_1, b_2] = \max_{\sigma_1} V^{\sigma_1|b_2}(b_1)\ . \tag{4}$$

As all $V^{\sigma_1|b_2}$ are linear, V^* is convex in b_1. Vice versa, for given fixed b_1, player 2 chooses a minimizing strategy,

$$V^*[b_1, b_2] = \min_{\sigma_2} V^{\sigma_2|b_1}(b_2)\ , \tag{5}$$

and V^* is concave in b_2. Since V^* is a pointwise maximum/minimum (Equations (4) and (5)) from $(U - L)$-Lipschitz continuous functions $V^{\sigma_i|b_{-i}}$, V^* is $(U - L)$-Lipschitz continuous in the dimension of b_1, as well as b_2. Combining the Lipschitz constants in these two dimensions results in $\sqrt{2} \cdot (U - L)$-Lipschitz continuity of V^*. □

Properties of Nash Equilibrium of PO-POSGs

Consider a Nash equilibrium strategy profile (σ_1, σ_2) and let $\pi_i(\cdot|s_i) = \sigma_i(s_i)$. If observations (o_1, o_2) are generated, the probability of transitioning to the joint state (s'_1, s'_2) is $\tau_{\pi_1}(b_2 o_2 o_1)(s'_1) \cdot \tau_{\pi_2}(b_1 o_1 o_2)(s'_2)$. Since the dynamics of the game is Markovian, the equilibrial strategies aim to optimize the payoff in the subgame after (o_1, o_2) is seen by the players—i.e., the expected discounted sum of the rewards starting from the joint belief $(\tau_{\pi_2}(b_1 o_1 o_2)(s'_2), \tau_{\pi_1}(b_2 o_2 o_1)(s'_1))$. Since in the equilibrium both players know this distribution, this expectation is equal to $V[\tau_{\pi_2}(b_1 o_1 o_2), \tau_{\pi_1}(b_2 o_2 o_1)]$ (since both players have strategies that guarantee this expected long-term reward when starting from the given joint belief).

This fact makes it possible to express the value of the equilibrial strategy profile (σ_1, σ_2) in terms of the immediate reward (direct consequences of the decisions in the first stage of the game) $R_{\pi_1\pi_2}$,

$$R_{\pi_1\pi_2} = \sum_{s_1, s_2, a_1, a_2} b_2(s_1) b_1(s_2) \pi_1(a_1|s_1) \pi_2(a_2|s_2) R(s_1 s_2 a_1 a_2) \tag{6}$$

and the values $V[\tau_{\pi_2}(b_1 o_1 o_2), \tau_{\pi_1}(b_2 o_2 o_1)]$ of the subgames:

$$R_{\pi_1\pi_2} + \tag{7}$$

$$+ \gamma \sum_{o_1, o_2} \Pr_{\pi_1\pi_2}[o_1 o_2] \cdot V[\tau_{\pi_2}(b_1 o_1 o_2), \tau_{\pi_1}(b_2 o_2 o_1)]\ .$$

Relaxing the assumption of known equilibrial strategies and performing the maximin optimization over Equation (7) gives us the value of the game starting in the joint belief (b_1, b_2) as a fixpoint equation over value functions

$$V^*[b_1,b_2] = HV^*[b_1,b_2] = \max_{\pi_1}\min_{\pi_2}\Big[R_{\pi_1\pi_2} + \tag{8}$$
$$+\gamma\sum_{o_1,o_2}\Pr_{\pi_1\pi_2}[o_1o_2]\cdot V[\tau_{\pi_2}(b_1o_1o_2),\tau_{\pi_1}(b_2o_2o_1)]\Big]\,.$$

Moreover, since $\gamma < 1$, the operator H defined over value functions $V : \Delta(S_2)\times\Delta(S_1)\to\mathbb{R}$ is a contraction. The Equation (8) can thus be used to approximate V^* iteratively.

Algorithm

Evaluating the dynamic programming operator H directly (as defined in Equation (8)) is impossible since the set of all joint beliefs is infinite. To design a practical algorithm, we need to establish an approximation scheme for V^* that we describe in this section first. Then we provide mathematical programs for computing HV when this approximation scheme is used. Finally, we state our algorithm to obtain ϵ-approximation of V^* in PO-POSGs.

Approximating V^*

In POMDPs (or one-sided POSGs), the value function V^* is commonly represented either as a point-wise maximum over a set of linear functions (termed α-vectors) or by considering a convex hull of a set of points. Both of these approaches leverage that the value function V^* is convex, which is not the case for PO-POSGs. In this section, we present a way to form a lower bound approximation $\underline{V}$ of a convex-concave function V^* inspired by both of the approaches mentioned above (the construction of the upper bound $\overline{V}$ is analogous).

To represent the value of $\underline{V}$ in the dimension of S_2, we use an extended notion of α-vectors, termed $\alpha\beta$-vectors. In PO-POSGs, the linear value $V^{\sigma_1|b_2}$ of a strategy depends on the belief b_2 of the adversary (see Lemma 1). Hence, also our $\alpha\beta$-vectors depend on the belief of the adversary denoted β. An $\alpha\beta$-vector consists of two components (see the thick line in Figure 1). First, there is a linear function $\alpha : \Delta(S_2)\to\mathbb{R}$ representing the value of the $\alpha\beta$-vector in the $\Delta(S_2)$ dimension. Second, there is a belief of the adversary $\beta\in\Delta(S_2)$ which informally positions the $\alpha\beta$-vector in the $\Delta(S_1)$ dimension. As a simplification, an $\alpha\beta$-vector can be seen as a value $V^{\sigma_1|b_2}$ of a strategy σ_1 in belief b_2, where $\alpha = V^{\sigma_1|b_2}$ and $\beta = b_2$. However, an $\alpha\beta$-vector of player 1 is an arbitrary function that lower bounds V^* in general.

Definition 5. An $\alpha\beta$-vector of player i is a tuple consisting of a linear function $\alpha : \Delta(S_{-i})\to\mathbb{R}$ and the belief of the adversary $\beta\in\Delta(S_i)$ satisfying

$$\alpha(b_1)\le V^*[b_1,\beta] \quad \text{, or} \quad \alpha(b_2)\ge V^*[\beta,b_2] \tag{9}$$

for player 1 or player 2, respectively. The set of all $\alpha\beta$-vectors of player 1 (player 2) used to construct the approximating function $\underline{V}$ ($\overline{V}$) is denoted Γ_1 (Γ_2, respectively).

Since V^* is concave in the belief of player 2 (i.e., $\Delta(S_1)$), every convex combination of $\alpha\beta$-vectors in Γ_1 forms a lower bound on V^*, and for every coefficients of a convex combination $\lambda(\alpha\beta)\ge 0$ satisfying $\sum_{\alpha\beta\in\Gamma_1}\lambda(\alpha\beta) = 1$,

$$\alpha'(b_1) = \sum_{\alpha\beta\in\Gamma_1}\lambda(\alpha\beta)\alpha(b_1), \beta' = \sum_{\alpha\beta\in\Gamma_1}\lambda(\alpha\beta)\beta \tag{10}$$

$\alpha'\beta'$ is also an (implicit) $\alpha\beta$-vector. The implicit $\alpha\beta$-vectors form facets in Figure 1.

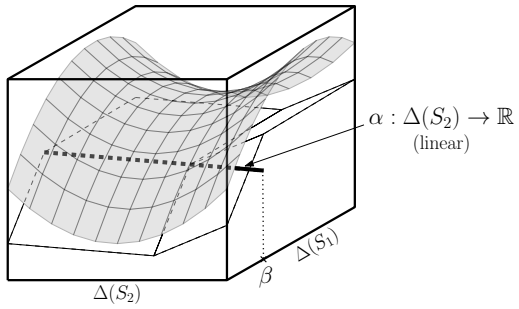

Figure 1: Lower bound on V^* using $\alpha\beta$-vectors of player 1.

Now, we leverage the α-vector representation of value functions as commonly used in POMDPs (or one-sided POSGs). To obtain the value $\underline{V}[b_1,b_2]$, a point-wise maximum over all (implicit) $\alpha\beta$-vectors with $\beta = b_2$ is taken.

$$\underline{V}[b_1,b_2] = \max_{\lambda(\cdot)\ge 0}\Big\{\sum_{\alpha\beta\in\Gamma_1}\lambda(\alpha\beta)\alpha(b_1)\;\Big|\;\sum_{\alpha\beta\in\Gamma_1}\lambda(\alpha\beta)\beta = b_2\Big\} \tag{11}$$

The upper-bounding value function $\overline{V}$ is constructed by considering $\alpha\beta$-vectors Γ_2 of player 2 and using a point-wise minimum instead of maximum. Lower and upper-bound approximations define the approximation error.

Definition 6. Let $\underline{V}$ and $\overline{V}$ be the current approximations of V^*. The approximation error (*gap*) in joint belief (b_1, b_2) is

$$\hat{V}[b_1,b_2] = \overline{V}[b_1,b_2] - \underline{V}[b_1,b_2]\,. \tag{12}$$

Computing $HV[b_1,b_2]$ via linear programming

When considering approximate value functions $\underline{V}$ and $\overline{V}$ described in the previous section, the point-based value backup $H\underline{V}[b_1,b_2]$ (or $H\overline{V}[b_1,b_2]$) can be evaluated using linear programming. We again focus on the construction of a linear program for computing lower bound $H\underline{V}[b_1,b_2]$ (denoted LP($H\underline{V}[b_1,b_2]$)), the case for LP($H\overline{V}[b_1,b_2]$) is analogous.

We start by rewriting the optimization problem $H\underline{V}[b_1,b_2]$ (as defined in Equation (7)) by evaluating $\underline{V}$ according to the Equation (11).

$$\max_{\pi_1,\lambda}\;\min_{\pi_2}\Big[R_{\pi_1\pi_2} + \gamma\sum_{o_1,o_2}\Pr_{\pi_2}[o_1]\cdot\Pr_{\pi_1}[o_2]\cdot \tag{13a}$$
$$\cdot\sum_{s_2'}\tau_{\pi_2}(b_1o_1o_2)(s_2')\sum_{\alpha\beta\in\Gamma_1}\lambda^{o_1o_2}(\alpha\beta)\cdot\alpha(s_2')\Big]$$
$$\text{s.t.}\sum_{\alpha\beta\in\Gamma_1}\lambda^{o_1o_2}(\alpha\beta)\cdot\beta(s_1') = \tau_{\pi_1}(b_2o_2o_1)(s_1') \tag{13b}$$
$$\forall(o_1,o_2)\in O_1\times O_2\;\;\forall s_1'\in S_1$$
$$\lambda(\cdot)\ge 0 \tag{13c}$$

Variables $\lambda(\cdot)$ from Equation (11) have been replaced with $\lambda^{o_1 o_2}(\cdot)$ for each observation pair. The term $1/Pr_{\pi_2}[o_1]$ in $\tau_{\pi_2}(b_1 o_1 o_2)$ cancels out, hence the objective becomes

$$\max_{\pi_1,\lambda} \min_{\pi_2} \Big[R_{\pi_1\pi_2} + \gamma \sum_{o_1,o_2} \Pr_{\pi_1}[o_2] \sum_{s_2,a_2,s_2'} b_1(s_2)\pi_2(a_2|s_2) \cdot \cdot Z(o_1|s_2 a_2) T(s_2'|s_2 a_2 o_2 o_1) \sum_{\alpha\beta\in\Gamma_1} \lambda^{o_1 o_2}(\alpha\beta)\alpha(s_2') \Big] . \quad (14)$$

Similarly, it is possible to cancel out $\Pr_{\pi_1}[o_2]$ in $\tau_{\pi_1}(b_2 o_2 o_1)$ by substituting $\hat{\lambda}^{o_1 o_2}(\cdot) = \Pr_{\pi_1}[o_2] \cdot \lambda^{o_1 o_2}(\cdot)$.

$$\max_{\pi_1,\hat{\lambda}} \min_{\pi_2} \Big[R_{\pi_1\pi_2} + \gamma \sum_{s_2,a_2,o_1,o_2,s_2'} b_1(s_2)\pi_2(a_2|s_2) Z[o_1|s_2 a_2] \cdot \cdot T[s_2'|s_2 a_2 o_2 o_1] \sum_{\alpha\beta\in\Gamma_1} \hat{\lambda}^{o_1 o_2}(\alpha\beta) \cdot \alpha(s_2') \Big] \quad (15a)$$

$$\text{s.t.} \sum_{\alpha\beta\in\Gamma_1} \hat{\lambda}^{o_1 o_2}(\alpha\beta) \cdot \beta(s_1') = \sum_{s_1,a_1} b_2(s_1)\pi_1(a_1|s_1) \cdot \cdot Z(o_2|s_1 a_1) T(s_1'|s_1 a_1 o_1 o_2) \quad \forall (o_1,o_2) \forall s_1' \quad (15b)$$

$$\hat{\lambda}(\cdot) \geq 0 \quad (15c)$$

When π_1 and $\hat{\lambda}$ variables are fixed, the value is linear in π_2. Hence the optimum will be in a pure strategy π_2. The minimization over a finite number of pure strategies can be rewritten using a set of linear inequality constraints. Moreover, we leverage the fact that the adversary (player 2) knows his current state. Therefore, it is possible to compute $\pi_2(\cdot|s_2)$ for each state s_2 of player 2 separately and compute the expectation over values of individual states. The resulting linear program follows.

$$\max_{\pi_1,\hat{\lambda}} \sum_{s_2} b_1(s_2) \cdot V(s_2)$$

$$\text{s.t. } V(s_2) \leq \sum_{s_1,a_1} b_2(s_1)\pi_1(a_1|s_1) R(s_1 s_2 a_1 a_2) + \quad (16a)$$
$$+ \gamma \sum_{o_1,o_2,s_2'} Z(o_1|s_2 a_2) T(s_2'|s_2 a_2 o_2 o_1) \cdot$$
$$\cdot \sum_{\alpha\beta\in\Gamma_1} \hat{\lambda}^{o_1 o_2}(\alpha\beta) \cdot \alpha(s_2') \qquad \forall s_2, a_2$$

$$\sum_{\alpha\beta\in\Gamma_1} \hat{\lambda}^{o_1 o_2}(\alpha\beta) \cdot \beta(s_1') = \sum_{s_1,a_1} b_2(s_1)\pi_1(a_1|s_1) \cdot \quad (16b)$$
$$\cdot Z(o_2|s_1 a_1) T(s_1'|s_1 a_1 o_1 o_2) \qquad \forall (o_1,o_2) \forall s_1'$$

$$\hat{\lambda}(\cdot) \geq 0 \quad (16c)$$

Note that the variables $V(s_2)$ correspond to the values of playing a strategy represented by values of variables π_1 and $\hat{\lambda}$ in the unobserved state s_2 of the opponent. Such strategy prescribes player 1 to play according to strategy π_1 in the first stage of the game and then, after observing (o_1, o_2), follow a strategy the value of which is greater than the convex combination of $\alpha\beta$-vectors with coefficients $\lambda(\alpha\beta)$,

$$\lambda(\alpha\beta) = \hat{\lambda}^{o_1 o_2}(\alpha\beta) / \sum_{\alpha\beta\in\Gamma_1} \hat{\lambda}^{o_1 o_2}(\alpha\beta) . \quad (17)$$

Hence, we can use values of the variables $V(s_2)$ to form a new $\alpha\beta$-vector ($\beta = b_2$) such that $\alpha(b_1) = \sum_{s_2} b_1(s_2) \cdot V(s_2)$. For states s_2 with $b_1(s_2) = 0$, the value $V(s_2)$ may, however underestimate, due to the lack of pressure on $V(s_2)$. In these cases, we compute the minimum represented by constraints (16a) separately.

The algorithm

We are now ready to state our algorithm to compute an ϵ-approximation of V^* in the joint belief $(b_1^{(0)}, b_2^{(0)})$ and to prove its correctness. The algorithm (Algorithm 1) follows the ideas of the HSVI algorithm for POMDPs (Smith and Simmons 2004) and one-sided POSGs (Horak, Bosansky, and Pechoucek 2017) while replacing the point-based update step with the computation of optimal $\alpha\beta$-vectors to add using the linear program from Equations (16).

1 Initialize $\underline{V}$ and $\overline{V}$
2 **while** $\hat{V}[b_1^{(0)}, b_2^{(0)}] > \epsilon$ **do** `explore` $(b_1^{(0)}, b_2^{(0)}, 0)$
3 **procedure** `explore` (b_1, b_2, t)
4 **if** $\hat{V}[b_1, b_2] \leq \rho(t)$ **then return**
5 Extract $\overline{\pi}_1$ from LP($H\overline{V}[b_1, b_2]$) and $\underline{\pi}_2$ from LP($H\underline{V}[b_1, b_2]$)
6 $(o_1^*, o_2^*) \leftarrow \arg\max_{o_1,o_2} Pr_{\overline{\pi}_1\underline{\pi}_2}[o_1 o_2] \cdot \text{excess}^{t+1}(\tau_{\underline{\pi}_2}(b_1 o_1 o_2), \tau_{\overline{\pi}_1}(b_2 o_2 o_1))$
7 `explore` $(\tau_{\underline{\pi}_2}(b_1 o_1 o_2), \tau_{\overline{\pi}_1}(b_2 o_2 o_1), t+1)$
8 Extract α_1 from LP($H\underline{V}[b_1, b_2]$) ($V(s_2)$ variables)
9 Extract α_2 from LP($H\overline{V}[b_1, b_2]$) ($V(s_1)$ variables)
10 $\Gamma_1 \leftarrow \Gamma_1 \cup \{\alpha_1 b_2\}$; $\Gamma_2 \leftarrow \Gamma_2 \cup \{\alpha_2 b_1\}$

Algorithm 1: HSVI algorithm for PO-POSGs.

Since we want to focus on the key characteristics of the algorithm, we initialize $\underline{V}$ and $\overline{V}$ using the minimum and maximum possible utilities of player 1,

$$L = \min_{s_1,s_2,a_1,a_2} R(s_1 s_2 a_1 a_2)/(1-\gamma) \quad (18)$$

$$U = \max_{s_1,s_2,a_1,a_2} R(s_1 s_2 a_1 a_2)/(1-\gamma) . \quad (19)$$

In practice, we can obtain tighter bounds (and consequently faster convergence) by either leveraging domain knowledge, or solving a simplified version of the game.

To obtain an ϵ-approximation of $V^*[b_1^{(0)}, b_2^{(0)}]$, it is sufficient that beliefs (b_1, b_2) reached at depth t (the value of which is therefore multiplied by γ^t) satisfy $\hat{V}[b_1, b_2] \leq \rho(t)$, where $\rho(t)$ is an increasing and unbounded sequence (for sufficiently small $R > 0$),

$$\rho(t) = \epsilon\gamma^{-t} - \sum_{i=1}^{t} 2R(U-L)\sqrt{2} \cdot \gamma^{-i} . \quad (20)$$

If $\hat{V}[b_1, b_2] > \rho(t)$, we say that (b_1, b_2) has a positive excess gap $\text{excess}^t(b_1, b_2) = \hat{V}[b_1, b_2] - \rho(t)$.

Once Algorithm 1 terminates, an ϵ-approximation of $V^*[b_1^{(0)}, b_2^{(0)}]$ has been found (see line 2). Moreover, since the sequence $\rho(t)$ is increasing and unbounded while the

maximum gap is bounded by $U - L$, the condition on line 4 is always eventually met and every call to `explore` therefore terminates in a bounded number of recursion levels (denote this bound $T_{\max}$). It is therefore sufficient to show that the number of calls to `explore` is finite.

Denote $\{(b_1^{(t)}, b_2^{(t)})\}_{t=0}^{T}$ the beliefs that have been visited during a trial of length T. Observe that $\hat{V}[b_1^{(T-1)}, b_2^{(T-1)}] > \rho(T-1)$ (otherwise the trial would have terminated at depth $(T-1)$). On the contrary, when considering belief $(b_1^{(T-1)}, b_2^{(T-1)})$ and the corresponding strategy profile $(\overline{\pi}_1, \underline{\pi}_2)$ from line 5, the reachable beliefs satisfy $\hat{V}[\tau_{\underline{\pi}_2}(b_1^{(T-1)} o_1 o_2), \tau_{\overline{\pi}_1}(b_2^{(T-1)} o_2 o_1)] \leq \rho(T)$ for every (o_1, o_2) seen with positive probability.

Lemma 2. *Consider a trial $\{(b_1^{(t)}, b_2^{(t)})\}_{t=0}^{T}$ of length T and consider that point-based updates on lines 8–10 of Algorithm 1 have been performed. Then*

(1) $\hat{V}[b_1^{(T-1)}, b_2^{(T-1)}] \leq \rho(T-1) - 2R(U-L)\sqrt{2}$, and

(2) For every (b_1, b_2) satisfying $\|(b_1, b_2) - (b_1^{(T-1)}, b_2^{(T-1)})\|_2 \leq R$, it holds $\hat{V}[b_1, b_2] \leq \rho(T-1)$.

Proof (sketch). Observe that from the definition of the sequence $\rho(t)$ in Equation (20) it follows that

$$\gamma\rho(T) = \rho(T-1) - 2R(U-L)\sqrt{2}\,. \tag{21}$$

Moreover, the trial terminated at depth T. Therefore, all beliefs that can be reached from $(b_1^{(T-1)}, b_2^{(T-1)})$ when following $(\overline{\pi}_1, \underline{\pi}_1)$ from line 5 must satisfy

$$\hat{V}[\tau_{\underline{\pi}_2}(b_1^{(T-1)} o_1 o_2), \tau_{\overline{\pi}_1}(b_2^{(T-1)} o_2 o_1)] \leq \rho(T)\,. \tag{22}$$

Let $(\underline{\pi}_1, \underline{\pi}_2)$ (and $(\overline{\pi}_1, \overline{\pi}_2)$) be equilibrial strategy profiles in $H\underline{V}[b_1^{(T-1)}, b_2^{(T-1)}]$ (and $H\overline{V}[b_1^{(T-1)}, b_2^{(T-1)}]$, respectively) and denote $u^{V}(\pi_1, \pi_2)$ the utility of playing strategies (π_1, π_2) in $HV[b_1^{(T-1)}, b_2^{(T-1)}]$. By deviating from the equilibrium, the players can only worsen their utility. Hence,

$$\begin{aligned} u^{\underline{V}}(\overline{\pi}_1, \underline{\pi}_2) \leq u^{\underline{V}}(\underline{\pi}_1, \underline{\pi}_2) = H\underline{V}[b_1^{(T-1)}, b_2^{(T-1)}] \leq \\ \leq H\overline{V}[b_1^{(T-1)}, b_2^{(T-1)}] = u^{\overline{V}}(\overline{\pi}_1, \overline{\pi}_2) \leq u^{\overline{V}}(\overline{\pi}_1, \underline{\pi}_2)\,. \end{aligned} \tag{23}$$

Since the same strategy profile $(\overline{\pi}_1, \underline{\pi}_2)$ is considered in both $u^{\underline{V}}(\overline{\pi}_1, \underline{\pi}_2)$ and $u^{\overline{V}}(\overline{\pi}_1, \underline{\pi}_2)$, the difference satisfies

$$\begin{aligned} u^{\overline{V}}(\overline{\pi}_1, \underline{\pi}_2) - u^{\underline{V}}(\overline{\pi}_1, \underline{\pi}_2) = \gamma \textstyle\sum_{o_1 o_2} \Pr_{\overline{\pi}_1 \underline{\pi}_2}[o_1 o_2] \cdot \\ \cdot \hat{V}[\tau_{\underline{\pi}_2}(b_1^{(T-1)} o_1 o_2), \tau_{\overline{\pi}_1}(b_2^{(T-1)} o_2 o_1)]\,. \end{aligned} \tag{24}$$

The gap $\hat{V}[\tau_{\underline{\pi}_2}(b_1 o_1 o_2), \tau_{\overline{\pi}_1}(b_2 o_2 o_1)]$ of all beliefs reachable using $(\overline{\pi}_1, \underline{\pi}_2)$ is smaller than $\rho(T)$ and hence $u^{\overline{V}}(\overline{\pi}_1, \underline{\pi}_2) - u^{\underline{V}}(\overline{\pi}_1, \underline{\pi}_2) \leq \gamma\rho(T)$. The point-based update in $(b_1^{(T-1)}, b_2^{(T-1)})$ renders $\hat{V}[b_1^{(T-1)}, b_2^{(T-1)}] \leq \rho(T-1) - 2R(U-L)\sqrt{2}$ which concludes the proof of (1).

Now, since V^* is $(U–L)\sqrt{2}$-Lipschitz continuous (Theorem 2), it is possible to consider $(U–L)\sqrt{2}$-Lipschitz continuous approximations $\underline{V}$ and $\overline{V}$. Function $\hat{V} = \overline{V} - \underline{V}$ is then $2(U–L)\sqrt{2}$-Lipschitz continuous and therefore the value of any belief within the R-neighborhood of $(b_1^{(T-1)}, b_2^{(T-1)})$ cannot be higher than $\rho(T-1)$ which proves (2). □

We are now ready to prove the correctness of the algorithm by showing that it can only perform a finite number of trials of given length.

Theorem 3. *Algorithm 1 terminates with an ϵ-approximation of $V^*[b_1^{(0)}, b_2^{(0)}]$.*

Proof. Assume for the sake of contradiction that the algorithm does not terminate and generates an infinite number of `explore` trials. Since the length of a trial is bounded by a finite number $T_{\max}$, the number of trials of length T (for some $0 \leq T \leq T_{\max}$) must be infinite. It is impossible to fit an infinite number of belief points (b_1, b_2) satisfying $\|(b_1, b_2) - (b_1', b_2')\|_2 > R$ within $\Delta(S_1) \times \Delta(S_2)$. Hence there must be two trials of length T, $\{(b_{11}^{(t)}, b_{21}^{(t)})\}_{t=0}^{T}$ and $\{(b_{12}^{(t)}, b_{22}^{(t)})\}_{t=0}^{T}$, such that $\|(b_{11}^{(T-1)}, b_{21}^{(T-1)}) - (b_{12}^{(T-1)}, b_{22}^{(T-1)})\|_2 \leq R$. Without loss of generality, assume that $(b_{11}^{(T-1)}, b_{21}^{(T-1)})$ was visited the first. According to Lemma 2, the point-based update in $(b_{11}^{(T-1)}, b_{21}^{(T-1)})$ resulted in $\hat{V}[(b_{12}^{(T-1)}, b_{22}^{(T-1)})] \leq \rho(T-1)$—which contradicts that the condition on line 4 of Algorithm 1 has not been satisfied for $(b_{12}^{(T-1)}, b_{22}^{(T-1)})$ (and hence that $\{(b_{12}^{(t)}, b_{22}^{(t)})\}_{t=0}^{T}$ was a trial of length T). □

Implementation details

In this section, we provide some of the details on our practical implementation of the HSVI algorithm for PO-POSGs.

Pruning The number of $\alpha\beta$-vectors grows in the course of the algorithm, however, not all of the vectors are needed to represent $\underline{V}$ (or $\overline{V}$) accurately. To counteract this growth, we run a pruning procedure every time the size of Γ_i gets $1.5\times$ larger than after the pruning was last performed. An $\alpha\beta$-vector is pruned if there exists a convex combination of vectors in Γ_i that dominates it.

Lipschitz continuity The theoretical proof of the correctness relies on the fact that the approximating functions are $(U-L)\sqrt{2}$-Lipschitz continuous. While the extracted $\alpha\beta$-vectors from the linear programs are $(U–L)\sqrt{2}$-Lipschitz continuous, the implicitly computed convex/concave hull need not satisfy this property (and thus may potentially render the Lipschitz constant of $\underline{V}$ or $\overline{V}$ impossible to bound). While this issue can be fixed by computing a $(U-L)\sqrt{2}$-Lipschitz envelope of $\underline{V}$ or $\overline{V}$ by adding additional $\alpha\beta$-vectors to Γ_i, we omit this step in our implementation. The computation of the envelope significantly increases the number of $\alpha\beta$-vectors (and thus the number of expensive pruning steps) and our experimental results show that the algorithm converges in practice even when the assumption of boundedly Lipschitz-continuous approximations is relaxed.

Other We use the idea of modifying ϵ between iterations similarly to (Horak, Bosansky, and Pechoucek 2017). The ϵ_{imm} for the current iteration is obtained as $\epsilon_{\text{imm}} = \epsilon + 0.5(\hat{V}[b_1^{(0)}, b_2^{(0)}] - \epsilon)$. This allows the algorithm to perform shorter trials in the initial phases of the search (when the bounds do not provide accurate information about what parts of the belief space to target).

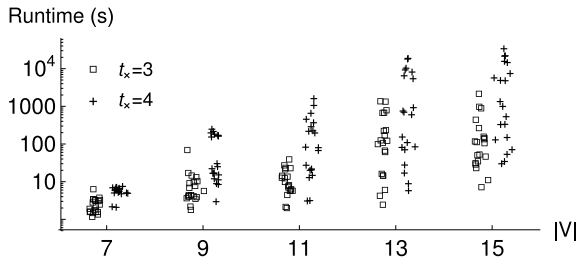

Figure 2: Experimental results on the Patrolling domain for different sizes of graph $|V|$. Time to reach $\hat{V}[b_1^{(0)}, b_2^{(0)}] \leq 1$.

We construct a compact version of the linear programs LP($HV[b_1, b_2]$). Namely, we consider only states, actions and observation pairs that can be played/observed in the current joint belief (b_1, b_2). Furthermore, we adopt a column generation approach to incrementally add variables $\lambda^{o_i o_j}(\cdot)$. Initially, we start with one $\alpha\beta$-vector (and its $\lambda^{o_i o_j}(\alpha\beta)$) for each pure belief of the opponent and we add additional $\alpha\beta$-vectors once they are necessary to accurately represent $V[\tau(b_1 o_1 o_2), \tau(b_2 o_2 o_1)]$.

Experiments

We demonstrate the scalability of our algorithm on two fundamentally different domains—partially observable patrolling inspired by (Basilico et al. 2009) and a lasertag game inspired by *Tag* from (Pineau, Gordon, and Thrun 2003). All experiments use discount factor $\gamma = 0.95$ and were run on Intel i7-8700K (solving 6 instances in parallel).

Patrolling The game is played by two players—the *patroller* and the *intruder*. The patroller moves between vertices V of a graph $G = (V, E)$ and attempts to locate an intruder before the intruder succeeds in causing damage. The intruder starts initially outside of the graph and observes the position of the patroller whenever he steps on one of the observable vertices $O \subseteq V$ (otherwise the position of the patroller remains hidden). The intruder may decide to attack any target vertex $v \in T, T \subseteq O$. Once the intruder decides to attack, he has to stay undetected in the chosen vertex v for $t_\times$ time steps to complete his attack and get a reward $c(v)$.

In our experimental evaluation, we consider $t_\times = 3$ and $t_\times = 4$ and generate random graphs from the Dorogovtsev-Mendes model such that the shortest cycle covering all targets is longer than $t_\times$ (i.e., the patroller cannot cover the targets perfectly). There are $|T| = \lceil V/4 \rceil$ targets and $|O| = \lceil 2V/3 \rceil$ observable nodes. The costs $c(v)$ of targets are generated uniformly from the $[70, 100]$ interval. Figure 2 summarizes the runtime of our algorithm on 200 randomly generated instances of Patrolling (time to reach precision 1, i.e., 1% of the maximum cost, is reported). All instances have been solved within 10 hours, while 97 instances with $t_\times = 3$ out of 100 and 82 instances with $t_\times = 4$ out of 100 have been solved in less than 20 minutes.

Lasertag The game is played by two players—the *tagger* and the *evader*—on a grid. In each time step, the players can decide to move to an adjacent square (free of an obstacle), or, the tagger can additionally shoot a laser beam either horizontally or vertically (which is effective until hitting the first obstacle). If the beam tags the evader, the tagger receives a reward $+10$ and the game ends, otherwise his reward is -10 and the game continues. Unless the tagger decides to use the laser beam, his reward is -1 in each step. Hence, the tagger attempts to terminate the game by tagging the adversary as quickly as possible. Neither player knows the position of each other until the tagger decides to shoot when the evader can observe the light ray (and thus deduce possible positions of the tagger).

We consider lasertag games played on a 4×4 grid with 3 obstacles where the tagger starts in the top-left corner, while the evader starts at position $(3, 4)$ next to the opposite corner. The obstacles are placed randomly while guaranteeing the existence of a path between the players (we discard symmetrical instances). We ran the algorithm with $\epsilon = 0.05$ for 5 hours. While the algorithm did not terminate within this limit on 16 out of 20 instances, the average excess gap in the initial belief relative to the value of the lower bound was $10\% \pm 2.6\%$ (where the confidence interval marks standard error). For grid size 3×3, all non-symmetric instances with players starting in opposite corners have been solved in less than 8 seconds.

Analysis We provide a detailed analysis of the performance of the algorithm for two instances of patrolling, an 11-vertex instance with $t_\times = 4$ solved in 307s and a 13-vertex instance with $t_\times = 4$ solved in 11004s. On both of the instances, 85% of the runtime corresponds to the operations with the approximating functions (especially computing values for a joint belief), while the construction and solving LP($HV[b_1, b_2]$) took only 10% of the runtime. The remaining 5% of the runtime corresponds to the pruning step, initiated 95 times on the larger instance within the 1556 iterations. The pruning eliminated 22126 $\alpha\beta$-vectors out of 50404 generated on the larger instance. Unlike in the patrolling domain, on a lasertag instance solved in 9337s the pruning was much more frequent (approximately one pruning per 6 iterations) and considerably more demanding (took 22% of runtime).

Conclusions

We present a subclass of partially observable stochastic games (POSGs) where the observations are publicly observable by the players. We provide the formal definition of such games and a novel, practical algorithm with proven convergence to approximately solve games in this class. Our algorithm is, to the best of our knowledge, the first practical general algorithm to solve a broad subclass of infinite-horizon games where both players lack information about the game state.

There is a large volume of possible future work. One direction is to adopt recent advancements from single-player POMDPs and apply them to the class of POSGs to improve scalability. Next, one can fine-tune representation of value functions and their initialization for specific games in security domains (e.g., in cybersecurity) to solve much larger

instances that correspond to real-world problems. Another way is to generalize the approach presented in this paper and relax some of the assumptions made: (1) adapt the approach for objective functions different from the discounted sum of rewards or (2) relax the factorization over the states.

Acknowledgments

This research was sponsored by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" and the Army Research Laboratory, and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Basilico, N.; Gatti, N.; Rossi, T.; Ceppi, S.; and Amigoni, F. 2009. Extending Algorithms for Mobile Robot Patrolling in the Presence of Adversaries to More Realistic Settings. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*.

Basilico, N.; Nittis, G. D.; and Gatti, N. 2016. A Security Game Combining Patrolling and Alarm–Triggered Responses Under Spatial and Detection Uncertainties. In *AAAI*.

Basu, A., and Stettner, L. 2015. Finite- and infinite-horizon shapley games with nonsymmetric partial observation. *SIAM Journal on Control and Optimization* 53(6):3584–3619.

Brazdil, T.; Kucera, A.; and Rehak, V. 2018. Solving Patrolling Problems in the Internet Environment. In *IJCAI*.

Brown, N.; Sandholm, T.; and Amos, B. 2018. Depth-Limited Solving for Imperfect-Information Games. In *NIPS*.

Chatterjee, K., and Doyen, L. 2014. Partial-observation stochastic games: How to win when belief fails. *ACM Transactions on Computational Logic* 15(2):16.

Chung, T. H.; Hollinger, G. A.; and Isler, V. 2011. Search and pursuit-evasion in mobile robotics. *Autonomous robots* 31(4):299–316.

Cole, H. L., and Kocherlakota, N. 2001. Dynamic games with hidden actions and hidden states. *Journal of Economic Theory* 98(1):114–126.

Durkota, K.; Lisý, V.; Kiekintveld, C.; Horák, K.; Bošanský, B.; and Pevný, T. 2017. Optimal Strategies for Detecting Data Exfiltration by Internal and External Attackers. In *GameSec*.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; and Lemieux, A. 2016. Deploying PAWS: Field optimization of the protection assistant for wildlife security. In *AAAI*.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *IJCAI*.

Ghosh, M. K.; McDonald, D.; and Sinha, S. 2004. Zero-Sum Stochastic Games with Partial Information. *Journal of Optimization Theory and Applications* 121(1):99–118.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *AAAI*.

Horak, K., and Bosansky, B. 2016. A Point-Based Approximate Algorithm for One-Sided Partially Observable Pursuit-Evasion Games. In *GameSec*.

Horak, K.; Bosansky, B.; and Pechoucek, M. 2017. Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games. In *AAAI*.

Kiekintveld, C.; Jain, M.; Tsai, J.; Pita, J.; Ordóñez, F.; and Tambe, M. 2009. Computing optimal randomized resource allocations for massive security games. In *AAMAS*.

Kumar, A., and Zilberstein, S. 2009. Dynamic programming approximations for partially observable stochastic games. In *FLAIRS*.

MacDermed, L. C. 2013. *Value Methods for Efficiently Solving Stochastic Games of Complete and Incomplete Information*. Ph.D. Dissertation, Georgia Institute of Technology.

Moravcik, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. *Science* 356(6337).

Nguyen, T. H.; Wellman, M. P.; and Singh, S. 2017. A Stackelberg Game Model for Botnet Data Exfiltration. In *GameSec*.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*.

Pita, J.; Jain, M.; Marecki, J.; Ordóñez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *AAMAS*.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; Direnzo, J.; Meyer, G.; Baldwin, C. W.; Maule, B. J.; and Meyer, G. R. 2012. PROTECT : A Deployed Game Theoretic System to Protect the Ports of the United States. In *AAMAS*.

Smith, T., and Simmons, R. 2004. Heuristic search value iteration for POMDPs. In *UAI*.

Vanek, O.; Yin, Z.; Jain, M.; Bosansky, B.; Tambe, M.; and Pechoucek, M. 2012. Game-theoretic Resource Allocation for Malicious Packet Detection in Computer Networks. In *AAMAS*.

von Neumann, J. 1928. Zur theorie der gesellschaftsspiele. *Mathematische Annalen* 100(1):295–320.

Vorobeychik, Y.; An, B.; Tambe, M.; and Singh, S. P. 2014. Computing Solutions in Infinite-Horizon Discounted Adversarial Patrolling Games. In *ICAPS*.