Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games

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

Abstract

Security problems can be modeled as two-player partially observable stochastic games with one-sided partial observability and infinite horizon (one-sided POSGs). We seek for optimal strategies of player 1 that correspond to robust strategies against the worst-case opponent (player 2) that is assumed to have a perfect information about the game. We present a novel algorithm for approximately solving one-sided POSGs based on the heuristic search value iteration (HSVI) for POMDPs. Our results include (1) theoretical properties of one-sided POSGs and their value functions, (2) guarantees showing the convergence of our algorithm to optimal strategies, and (3) practical demonstration of applicability and scalability of our algorithm on three different domains: pursuit-evasion, patrolling, and search games.

Introduction

Game theory is widely used in security problems and strategies from game-theoretic models are applied to protect critical infrastructures (Pita et al. 2008; Kiekintveld et al. 2009; Shieh et al. 2012), computer networks (Vanek et al. 2012) or wildlife (Fang, Stone, and Tambe 2015; Fang et al. 2016). Many real-world situations, however, contain a dynamic strategic interaction between the players that has to be addressed in the models. Players can observe (possibly imperfectly) information about actions of their opponent and react to these observations. Examples include patrolling games (Basilico, Gatti, and Amigoni 2009; Vorobeychik et al. 2014; Basilico, Nittis, and Gatti 2016), 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.

Finding optimal strategies in such dynamic games with imperfect information is often computationally challenging. If the horizon of the interaction is restricted, we can use the *extensive-form games* formulation. Typically, the size of this representation grows exponentially with the horizon and prohibits us from solving large games. If the horizon is infinite (or indefinite), we can use *partially observable stochastic games (POSGs)*. In POSGs, however, many problems are undecidable (Madani, Hanks, and Condon 1999) even when we use a discount factor to restrict future gains.

However, real-world security scenarios naturally require partial observability and no strictly defined horizon. The goal is to find best robust strategies that provide guarantees on the expected outcome for one player (the defender) against any opponent (the attacker). Therefore, we focus on *discounted two-player zero-sum POSGs with concurrent moves and one-sided partial observability* where it is assumed that the attacker has full information about the game – the attacker knows the state of the game as well as the history of actions played. One-sided partial observability has been used in specific domains such as patrolling games, e.g. (Vorobeychik et al. 2014), or pursuit-evasion games, e.g. (Horak and Bosansky 2016). We generalize this concept to a broad class of POSGs.

Our main contribution is the first domain-independent algorithm that has guarantees to approximate optimal strategies in one-sided POSGs. Our algorithm is a generalization of the heuristic search value iteration algorithm (HSVI) for Partially Observable Markov Decision Processes (POMDPs). Similarly to POMDPs, one-sided POSGs allow us to compactly represent strategies and value functions representing values of the game based on the belief the first player has about the state of the game. Contrary to POMDPs, the presence of the opponent player causes significant technical challenges that we address in this paper. First, we show that the assumption of the one-sided partial observability guarantees that the value functions are convex. Second, we define a value backup operator and show that an iterative application of this operator converges to the optimal values. Third, we generalize the ideas behind HSVI towards one-sided POSGs, and show that our algorithm approximates optimal strategies. Finally, we demonstrate the applicability and scalability of our algorithm on three different domains – patrolling games (including the variant with alarms), pursuit-evasion games, and search games. The results show that our algorithm can closely approximate solutions of large games with more than 4000 states.

Related Work

There are only a few relevant algorithms for computing strategies in POSGs. An algorithm for computing strategies in POSGs where all players have imperfect information was proposed in (Hansen, Bernstein, and Zilberstein 2004). The algorithm approximates an infinite horizon game by increas-

ing the horizon in a finite-horizon game and uses dynamic programming to incrementally construct a set of relevant pure strategies by eliminating dominated strategies. The set of such strategies is then used to form a normal-form (or matrix) representation of the POSG. However, the exponential transformation to the normal form prevents this algorithm from scaling up. One-sided partial observability allows us to avoid such enumeration of pure strategies.

The closest works related to the algorithm presented in this paper are two works on a specific subclass of one-sided POSGs – pursuit-evasion games (PEGs). First, a class of *one-sided partially observable PEGs* was presented and theoretical results on the shape of the value functions and the definition of the value backup operator were provided in (Horak and Bosansky 2017). Second, an HSVI-based algorithm was introduced in (Horak and Bosansky 2016).

Our algorithm can be seen as a significant generalization of this approach to a broader class of one-sided POSGs. First, the set of observations is very limited in PEGs – player 1 is able to observe his own actions only and the only direct information about the position of the opponent is given when player 2 is captured. Considering general observations presents additional challenges for the model and the algorithm which we address in this paper. Secondly, the previous work relied on a uniform sampling of belief points to guarantee the convergence, our algorithm approximates the solution in a deterministic manner.

Two-Player One-Sided POSGs

A *one-sided partially observable stochastic game* G is a tuple $G = \langle \mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, \mathcal{O}, \mathcal{T}, \mathcal{R} \rangle$. The game is played for an infinite number of *stages*. At each stage, the game is in one of the states $s \in \mathcal{S}$ and players choose their actions $a \in \mathcal{A}_1$ and $a' \in \mathcal{A}_2$ simultaneously. An initial state of the game is drawn from a probability distribution $b^0 \in \Delta(\mathcal{S})$, which we treat as a parameter of the game and term the *initial belief*.

The choice of actions determines the outcome of the current stage: Player 1 gets an *observation* $o \in \mathcal{O}$ and the game moves to a state $s' \in \mathcal{S}$ with probability $\mathcal{T}_{s,a,a'}(o, s')$, where s is the current state. Furthermore he gets a reward $\mathcal{R}(s, a, a')$ for this transition. We assume the zero-sum case, hence player 2 receives $-\mathcal{R}(s, a, a')$, and we assume that the rewards are discounted over time with discount factor $\gamma < 1$. Players do not observe their rewards during the game.

We assume perfect recall, hence both players remember their respective histories. A history of the first player is formed by actions he played and observations he received, i.e. $(\mathcal{A}_1 \times \mathcal{O})^t$. The second player has complete observation, hence $\mathcal{S} \times (\mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{O} \times \mathcal{S})^t$ is a set of her histories. The strategies σ_1, σ_2 of the players map each of their histories to a distribution over their actions.

Value of a Strategy and Value of the Game

In this section, we show that the value of a strategy (the expected reward of the first player playing σ_1 when the opponent plays her best response) has a linear dependence on the belief.

The value of the game G is the value of the best strategy available for each of the initial beliefs $b^0 \in \Delta(\mathcal{S})$. We represent the value of a game (based on the initial belief) as a *value function*. This function is a pointwise maximum taken over values of all strategies of the first player, which, since the value of every strategy is linear, forms a convex function.

In the convergence proof of our algorithm, we exploit that the rate of change in the value function is bounded in terms of minimum and maximum rewards of G, i.e. the value function is Lipschitz continuous.

Definition 1 (Value functions)**.** *The value of a strategy σ_1 of the first player is a function $v_{\sigma_1} : \Delta(\mathcal{S}) \to \mathbb{R}$ which assigns the expected utility $v_{\sigma_1}(b^0)$ of the player 1 in the game with initial belief b^0 when the first player follows σ_1 and the second player best-responds. The value function of the game G is a function $v^* : \Delta(\mathcal{S}) \to \mathbb{R}$ that assigns the value $v^*(b^0)$ of the best strategy of the first player for each of the beliefs, i.e. $v^*(b^0) = \sup_{\sigma_1} v_{\sigma_1}(b^0)$.*

Lemma 1. *The value v_{σ_1} of a fixed strategy σ_1 of the first player is linear in the initial belief.*

The proof relies on the fact that the player 2 knows the initial state of the game; hence, the initial belief forms a convex combination of values of best responses for individual states. Due to the space constraints, full proofs of all lemmas can be found in the full version of the paper.

We say that a function f is K-Lipschitz if it satisfies $|f(x) - f(y)| \leq K \cdot \|x - y\|_2$. The key observation to derive the Lipschitz continuity is that the value of the game lies in a bounded interval $[L, U]$ where

$$L = \min_{(s,a,a')} \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s, a, a') \,, \quad U = \max_{(s,a,a')} \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s, a, a') \,.$$

The proof of the following lemma then relies on defining the value of the strategy by assigning these extreme values to the vertices of the belief simplex and identifying the configuration with the largest rate of change.

Lemma 2. *Value function v_{σ_1} of a fixed strategy σ_1 of player 1 is $(U-L)$-Lipschitz.*

Theorem 1. *Value function v^* of the game G is convex in the initial belief and $(U-L)$-Lipschitz.*

Proof. The value function v^* is the supremum taken over a set of $(U-L)$-Lipschitz functions corresponding to values of strategies available to player 1 (Def. 1, Lemma 2). Supremum taken over a family of bounded $(U-L)$-Lipschitz continuous functions is $(U-L)$-Lipschitz continuous. Moreover since these functions are linear (Lemma 1), the resulting value function is convex. □

Value Backup

Now we present a value iteration algorithm for solving one-sided POSGs. The algorithm approximates the value function v^* of the infinite horizon game G by considering value

functions of the game with a restricted horizon. Each iteration of the algorithm improves the approximation by increasing the horizon by one step using the *value backup operator* (denoted H). Applying this operator means that players choose their Nash equilibrium strategies in the current step while assuming that the value of the subsequent stage is represented by the value function from the previous iteration.

The algorithm constructs a sequence $\{v^t\}_{t=0}^{\infty}$, starting with a value function v^0 of a game where only immediate rewards are considered. First, we discuss application of the operator in a single stage. Afterward we show the convergence when the operator is applied repeatedly.

Value Backup Operator

The value backup operator H evaluated at belief b — $[Hv](b)$— corresponds to solving a *stage game* where players choose their Nash equilibrium strategies for one stage of the game (in latter text we use $[Hv](b)$ to refer to this game as well). We denote strategies for one stage $\pi_1 \in \Delta(\mathcal{A}_1)$ for the first player and $\pi_2 : \mathcal{S} \to \Delta(\mathcal{A}_2)$ for the player 2. The utilities in $[Hv](b)$ depend both on the immediate rewards $\mathcal{R}$ and the discounted value of the subsequent game represented by value function v. The immediate rewards part depends solely on the actions played by the players:

$$R^{\text{imm}}_{\pi_1,\pi_2} = \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_1} \sum_{a'\in\mathcal{A}_2} b(s)\cdot\pi_1(a)\cdot\pi_2(s,a')\cdot\mathcal{R}(s,a,a') \quad (1)$$

The first player both knows the action a he played and observes observation o. He can use this information to derive his belief for the subsequent game:

$$b^{a,o}_{\pi_2}(s') = \frac{1}{\Pr[o|a,\pi_2]} \sum_{s\in\mathcal{S}} \sum_{a'\in\mathcal{A}_2} \mathcal{T}_{s,a,a'}(o,s')\cdot b(s)\cdot\pi_2(s,a') \quad (2)$$

The value of the subsequent game is then the expectation taken over individual action-observation pairs (a,o) of the first player from the values of a game starting in belief $b^{a,o}_{\pi_2}$:

$$R^{\text{subs}}_{\pi_1,\pi_2}(v) = \sum_{a\in\mathcal{A}_1} \sum_{o\in\mathcal{O}} \pi_1(a)\cdot\Pr[o|a,\pi_2]\cdot v(b^{a,o}_{\pi_2}) \,. \quad (3)$$

Since the value function is convex, utility of playing strategy profile (π_1,π_2) is convex when π_1 is fixed and linear when we fix π_2. The minimax theorem (von Neumann 1928; Nikaido 1954) applies and the Nash equilibrium strategy is solved by maximin/minimax:

$$[Hv](b) = \min_{\pi_2}\max_{\pi_1}\left(R^{\text{imm}}_{\pi_1,\pi_2} + \gamma\cdot R^{\text{subs}}_{\pi_1,\pi_2}(v)\right) \,. \quad (4)$$

Computation of Value Backup Operator

Finally, we present the way of computing $[Hv](b)$. When the value function v is piecewise linear and convex (PWLC), it can be represented by a set Γ of α-vectors and the value backup $[Hv](b)$ can be evaluated by means of linear programming. Each α-vector $\alpha\in\Gamma$ is an $|\mathcal{S}|$-tuple representing the affine value function v_{σ_1} of a fixed strategy σ_1 by specifying its values in each of the pure beliefs ($\alpha(s)$ for each $s\in\mathcal{S}$). We focus on the problem of solving the problem from the perspective of the second player first, who has to choose her strategy π_2 such that the utility V of the best responding player 1 (who chooses his pure best response $a\in\mathcal{A}_1$) is minimized.

The value of playing strategy π_2 against action $a\in\mathcal{A}_1$ equals $R^{\text{imm}}_{a,\pi_2} + \gamma R^{\text{succ}}_{a,\pi_2}(v)$, which allows us to construct a set of best-response constraints (one for each action a)

$$V \geq \sum_{s\in\mathcal{S}} \sum_{a'\in\mathcal{A}_2} b(s)\cdot\pi_2(s,a')\cdot\mathcal{R}(s,a,a') + \\ + \gamma\sum_{o\in\mathcal{O}} \Pr[o|a,\pi_2]\cdot v(b^{a,o}_{\pi_2}) \,. \quad (5)$$

Assuming that the value function v is represented by a set Γ of α-vectors, such that $v(b) = \max_{\alpha\in\Gamma}\langle\alpha,b\rangle$ ($\langle\cdot,\cdot\rangle$ denotes an inner product), its value can be rewritten by a set of inequalities

$$v(b^{a,o}_{\pi_2}) \geq \sum_{s'\in\mathcal{S}} \alpha(s')\cdot b^{a,o}_{\pi_2}(s') \qquad \forall\alpha\in\Gamma \quad (6)$$

where $b^{a,o}_{\pi_2}(s')$ is represented by linear constraints corresponding to Eq. (2). The term $\Pr[o|a,\pi_2]$ occuring in Eqs. (2) and (5) cancels out to form the resulting linear program.

Strategy of the First Player One way to approximate the value function by a PWLC function is to use a finite subset of strategies of the first player. Value functions of these strategies are linear (Lemma 1), and the pointwise maximum from these linear functions gives us the desired PWLC approximation. In such a case, each of the vectors in Γ corresponds to the value function of one of the strategies. The dual linear program is used to find the optimal control strategy of the first player, when duals of Eq. (5) correspond to the strategy to play in the first stage (when the history of the first player is empty) and duals of Eq. (6) prescribes what strategy to follow when (a,o) was observed in the first stage.

Convergence of the Value Backup Operator

In this section we show that a repetitive application of the value backup operator H converges to the *same* value function v^* of the infinite horizon game regardless of what value function it is applied on. We show this by demonstrating that the operator H is a contraction mapping with a factor $\gamma<1$.

Lemma 3. *Let v, v' be value functions, $b\in\Delta(\mathcal{S})$ be a belief and π_1, π_2 (resp. π_1', π_2') be equilibrial strategies in $[Hv](b)$ (resp. $[Hv'](b)$). Assume that for every action-observation pair (a,o) of the first player, $|v(b^{a,o}_{\pi_2}) - v'(b^{a,o}_{\pi_2})| \leq \mu$. Then $|[Hv](b) - [Hv'](b)| \leq \gamma\mu$.*

The lemma is proven by modyfing Nash equilibrium strategy profiles in games $[Hv](b)$ and $[Hv'](b)$ and bounding the difference by the difference of their expected utilities.

Theorem 2. *The operator H is a contraction mapping under the norm $\|v - v'\| = \max_{b\in\Delta(\mathcal{S})}|v(b) - v'(b)|$. It thus has a unique fixpoint – the value function of the infinite horizon game.*

Data: Game $\langle \mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, \mathcal{O}, \mathcal{T}, \mathcal{R} \rangle$, initial belief b^0, discount factor γ, desired precision $\epsilon > 0$, neighborhood parameter R
Result: Approximate value function $\hat{v}$
1 Initialize $\hat{v}$
2 **while** $\mathrm{gap}(\hat{v}(b^0)) > \epsilon$ **do**
3 | `Explore` $(b^0, \epsilon, R, 0)$
4 **return** $\hat{v}$
5 **procedure** `Explore` (b, ϵ, R, t)
6 | $\pi_2 \leftarrow$ optimal strategy of player 2 in $[H\underline{v}](b)$
7 | $(a, o) \leftarrow$ select according to forward exploration heuristic
8 | **if** $\mathrm{excess}(\hat{v}(b^{a,o}_{\pi_2}), t+1) > 0$ **then**
9 | | `Explore` $(b^{a,o}_{\pi_2}, \epsilon, R, t+1)$
10 | $\Gamma \leftarrow \Gamma \cup \{L\Gamma(b)\}$
11 | $\Upsilon \leftarrow \Upsilon \cup \{U\Upsilon(b)\}$ and make $\overline{v}$ $(U-L)$-Lipschitz
Algorithm 1: HSVI algorithm for one-sided POSGs

Proof. Let $\|v - v'\| \leq \mu$. Then for every $b^{a,o}_{\pi_2}$ from Lemma 3 $|v(b^{a,o}_{\pi_2}) - v'(b^{a,o}_{\pi_2})| \leq \mu$ and for every belief b, $|[Hv](b) - [Hv'](b)| \leq \gamma\mu$. The uniqueness of the fixpoint and the convergence properties follow from the Banach's fixed point theorem (Ciesielski 2007). □

HSVI Algorithm for POSGs

Similarly to POMDPs, the value iteration algorithm cannot scale for practical problems. We thus present a point-based algorithm (Algorithm 1) that by sampling the belief space bounds and approximates the true value function v^* of the game by a pair of PWLC functions $\underline{v}$ (*lower bound*), represented by a set of α-vectors Γ, and $\overline{v}$ (*upper bound*) represented as a lower envelope of a set of points Υ. We refer to these functions jointly as $\hat{v}$. The goal of the algorithm is to ensure that the *gap* in the initial belief b^0 of the game induced by the approximation defined as $\mathrm{gap}(\hat{v}(b)) = \overline{v}(b) - \underline{v}(b)$ is no higher than the required precision. Functions $\hat{v}$ are refined by adding new elements to their sets. These new elements result from *point-based updates* of operator H at a single belief point b.

The algorithm is initialized with $\underline{v}$ (and Γ) corresponding to the value of a uniform strategy of the first player and the upper bound $\overline{v}$ (and Υ) results from solving a perfect information refinement of the game. In every iteration, a finite set of beliefs is updated by *forward exploration* (lines 6-9). Beliefs selected by this process contribute to the fact that the gap at b^0 is not sufficiently small, and hence the approximation in these beliefs needs to be improved by applying point-based updates (lines 10 and 11). We now describe how the updates are performed, followed by the description of the forward exploration search.

Point-Based Updates

A point-based update at belief point b updates the lower and upper bound functions $\underline{v}$ and $\overline{v}$ using the optimal strategies in games $[H\underline{v}](b)$ and $[H\overline{v}](b)$. In order to prove the convergence, we require that the functions $\underline{v}$ and $\overline{v}$ are $(U-L)$-Lipschitz; hence, the update has to preserve this property.

The update of $\underline{v}$ adds an α-vector corresponding to the value of a Nash equilibrium strategy of the first player in $[H\underline{v}](b)$ (denoted $L\Gamma(b)$) computed from duals of the linear program (Eqs. (5)-(6)). The value of such strategy is linear and $(U-L)$-Lipschitz (Lemma 2), hence the expansion of Γ by $L\Gamma(b)$ preserves $(U-L)$-Lipschitz continuity of $\underline{v}$.

The upper bound function $\overline{v}$ is represented by a set of points Υ. Update of upper bound adds one point, $U\Upsilon(b) = b \to [H\overline{v}](b)$, that corresponds to the evaluation of the value backup at belief b. We cannot use the linear program outlined in Eqs. (5)-(6) to compute $[H\overline{v}](b)$ directly since the function $\overline{v}$ is not represented using α-vectors. We, therefore, use a transformation presented in (Horak and Bosansky 2016) which performs projections of beliefs to the lower envelope of $\overline{v}$ while preserving linearity of the constraints.

Adding a point to Υ can break the $(U-L)$-Lipschitz continuity of $\overline{v}$. We can fix this by constructing a piecewise linear approximation of a lower $(U-L)$-Lipschitz envelope:

$$\overline{v}(b) \coloneqq \inf_{b' \in \Upsilon} \left\{ \overline{v}(b') + (U-L) \cdot \|b - b'\|_2 \right\} . \tag{7}$$

The resulting function is $c(U-L)$-Lipschitz when c depends on the accuracy of the approximation and can be arbitrarily close to 1.

Forward Exploration

The value backup operator H expresses the value in belief b in terms of values of subsequent beliefs $b^{a,o}_{\pi_2}$. When applied to value functions $\hat{v}$, it also propagates the approximation error. In order to minimize the gap in the initial belief b^0, we need to achieve sufficient accuracy also in beliefs encountered at a later *time*.

The forward exploration simulates a play between the players while assuming that the second player follows a strategy obtained from the application of H on the lower bound $\underline{v}$ (i.e. she is overly optimistic with her strategy). When a belief b is encountered at time t (we term such a pair (b, t) a *timed belief*) and its approximation $\hat{v}(b)$ is not sufficiently accurate, we say that it has positive *excess gap*.

Definition 2 (Excess gap)**.** *Let ϵ be the desired precision and $R > 0$ be a neighborhood parameter. Let*

$$\rho(t) = \epsilon\gamma^{-t} - \sum_{i=1}^{t} 2R(U-L)\gamma^{-i} . \tag{8}$$

We define the excess gap of a timed belief (b, t) as

$$\mathrm{excess}(b, t) = \mathrm{gap}(\hat{v}(b)) - \rho(t) . \tag{9}$$

Later we show that if all subsequent timed beliefs $(b^{a,o}_{\pi_2}, t+1)$ have negative excess gap, a point-based update at (b, t) makes the excess gap $\mathrm{excess}(b, t)$ negative as well (in fact, $\mathrm{excess}(b, t) \leq -2R(U-L)$; we then term (b, t) as *closed*). If this does not hold for the belief (b, t) currently explored, the forward exploration process selects one of the subsequent beliefs $(b^{a,o}_{\pi_2}, t+1)$ with a positive excess gap for further exploration and the process is repeated with the timed belief $(b^{a,o}_{\pi_2}, t+1)$. If all subsequent beliefs have a negative excess gap, the forward exploration process terminates.

The termination is guaranteed if the neighborhood parameter R is chosen so that the sequence $\rho(t)$ is monotonically increasing in t and unbounded.

Forward Exploration Heuristic A positive excess gap of a belief contributes to the approximation error in the initial belief. If there are multiple subsequent timed beliefs $(b^{a,o}_{\pi_2}, t+1)$ with a positive excess gap, we select the one with the highest *weighted* excess gap which is similar to the weighted excess heuristic used in (Smith and Simmons 2004). The excess gap is weighted by both the observation probability *and* the probability that the first player plays a given action when using the strategy obtained from the upper bound value function $\overline{v}$ (i.e. according to the strategy π_1 from the game $[H\overline{v}](b)$). The action observation pair (a, o) selected in timed belief (b, t) for the further exploration maximizes $\pi_1(a) \cdot \Pr[o|a, \pi_2] \cdot \text{excess}(b^{a,o}_{\pi_2}, t+1)$.

Convergence of the Algorithm

The goal of the HSVI algorithm is to make the excess gap negative in all reachable timed beliefs and thus sufficiently decrease the gap in the initial belief. Contrary to POMDPs, reachable beliefs in POSGs are influenced by the strategy of the second player – she can change her strategy to reach a belief (b', t) with a positive excess gap instead of a closed belief (b, t), while b' stays arbitrarily close to b.

We avoid this by ensuring that if (b', t) with a positive excess gap is reached by the forward exploration, it lies sufficiently far from all previously closed beliefs at time t – the minimum distance between the beliefs being controlled by the neighborhood parameter $R > 0$ from the definition of the excess gap. Unlike in POMDPs, our modified definition of the excess gap ensures that not only a closed belief itself gets a negative excess gap: all beliefs within its R-neighborhood get a negative excess gap as well (Lemma 4). The convergence of the algorithm follows since there is only a finite number of such R-separated belief points.

Lemma 4. *Let (b, t) be a timed belief and π_2 be the optimal strategy of the second player in $[H\underline{v}](b)$. If $\text{excess}(b^{a,o}_{\pi_2}, t+1) \leq 0$ for all action-observation pairs (a, o) of the first player, then after performing a point-based update at b it holds that (i) $\text{excess}(b, t) \leq -2R(U-L)$ and (ii) all belief points b' in the R-neighborhood of b (i.e. $\|b - b'\|_2 \leq R$) have a negative excess gap $\text{excess}(b', t)$.*

The first part of the lemma follows from Lemma 3, the latter follows from $2(U-L)$-Lipschitz continuity of difference of $(U-L)$-Lipschitz functions $\underline{v}$ and $\overline{v}$.

Definition 3. *Let t be time. The set of all beliefs with negative excess gap at time t is denoted Ψ_t;*

$$\Psi_t = \{b \in \Delta(S) \mid \text{gap}(\hat{v}(b)) \leq \rho(t)\} \ . \tag{10}$$

Theorem 3. *HSVI algorithm converges to the precision ϵ.*

Proof. In each iteration, the algorithm performs a forward exploration until it encounters a timed belief (b, t) such that all subsequent timed beliefs $(b^{a,o}_{\pi_f}, t+1)$ have a negative excess gap. Since $\text{gap}(b^{a,o}_{\pi_f})$ is bounded by $U-L$, this happens after at most $t_{\max}$ steps, where

$$t_{\max} = \left\lceil \log_{1/\gamma} \left(\frac{U-L}{\epsilon} \cdot \left[1 + 2R \frac{1-\gamma^t}{\gamma^t(1-\gamma)} \right] \right) \right\rceil \ . \tag{11}$$

When the terminal timed belief (b, t) is reached, then $b \notin \Psi_t$ and all subsequent timed beliefs have negative excess gap. After performing the point-based update at (b, t), the excess gap of (b, t), as well as of all timed beliefs in the R-neighborhood of (b, t), is negative (Lemma 4) and Ψ_t is expanded. We show that the expansion of the sets $\Psi_{t'}$ guarantees that eventually $\Psi_{t'} = \Delta(\mathcal{S})$ for all times $t' \leq t_{\max}$, unless the desired precision ϵ is achieved beforehand.

The distance of b from the nearest belief b' in Ψ_t previously closed by the algorithm is at least R, since all points in the R-neighborhood of b' have a negative excess gap and thus are in Ψ_t. In each iteration, Ψ_t is expanded by at least one belief and (at least) its R-neighborhood.

The number of such expansions of timed beliefs is finite. In fact, the problem of finding maximum set of R-separated beliefs can be seen as a *hypersphere packing* (a higher dimensional version of the sphere packing (Hales 2011)) filling the belief simplex using non-overlapping hyperspheres of radius $R/2$, since the hyperspheres do not overlap exactly when the distance between their centers is at least R. When no hypersphere can be further inserted, it means that we cannot find any belief with a positive excess gap, hence we reached the desired precision in the whole belief space. □

Experiments

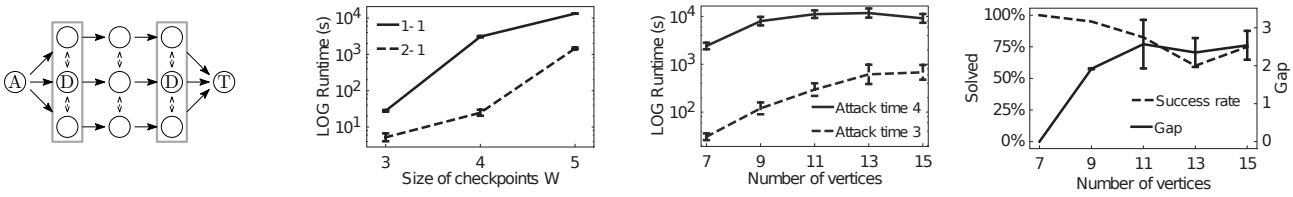
We demonstrate application possibilities and scalability of our algorithm on three types of games: pursuit-evasion games (e.g., evaluated in (Horak and Bosansky 2016)), intrusion search games (e.g., see (Bosansky et al. 2014)), and patrolling games with a discount factor (e.g., see (Vorobeychik et al. 2014)). Each player is assigned a team of units (either one or multiple units) located in vertices of a graph and he or she controls their movement on the graph. A move consists of moving the units simultaneously to vertices adjacent to their current positions, or they can wait.

The utilities are scaled so that the values of the games lies in the interval $[0, 100]$ (or $[-100, 0]$, respectively). Unless stated otherwise the discount factor is $\gamma = 0.95$ and we ran the algorithm until $\text{gap}(\hat{v}(b^0)) \leq 1$.

Algorithm Settings

We initialize the value functions by solving the perfect-information refinement of the game (for $\overline{v}$) and as a best response to a uniform strategy of player 1 (for $\underline{v}$). We use standard value iteration for stochastic games, or MDPs, respectively, and terminate the initialization when either change in valuations between iterations is lower than 0.025, or $20s$ time limit has expired. The initialization time is included in the computation times of the algorithms.

Similarly to (Smith and Simmons 2004), we adjust ϵ in each iteration using formula $\epsilon = 0.25 + \eta(\text{gap}(\hat{v}(b^0)) - 0.25)$ with $\eta = 0.9$. We set the neighborhood parameter R to the largest value satisfying $\rho(t) \geq 0.25\gamma^{-t}$ for all $t \leq t_{\max}$ from the proof of Theorem 3.



(a) (b) (c) (d)

Figure 1: (a) Intrusion-search game: $W = 3$, configuration 1-1: A denotes initial position of the attacker, D initial positions of defender's units, T is attacker's target (b) Intrusion-search games with 2 zones, each with W vertices: Time to reach $\mathrm{gap}(\hat{v}(b^0)) \leq 1$ (c) Patrolling games played on graphs generated from $ER(0.25)$: Time to reach $\mathrm{gap}(\hat{v}(b^0)) \leq 1$ (only successfully solved instances within 10 hours) (d) Patrolling games played on graphs generated from $ER(0.25)$: Percentage of successfully solved instances with $t_\times = 4$ and the gap on failed instances after 10 hours

Finally, we remove dominated points and vectors from sets Γ and Υ whenever their size grows by 20% to reduce the size of the linear programs. Again, this is similar to POMDPs (Smith and Simmons 2004).

Pursuit-Evasion Games (PEGs)

A team of centrally controlled pursuers aims to locate and capture the evader, and receive the utility of +100; the evader aims for the opposite. We consider $3 \times N$ grid graphs (we vary the number of columns N), two pursuing units start in top left positions, the evader starts in bottom right corner. Our algorithm achieves similar scalability as the existing algorithm designed specifically for one-sided PEGs (Horak and Bosansky 2016) and displays exponential dependence of the runtime on the width of the grid N. The game with $N = 3$ was solved in 9s on average, the game with $N = 6$ took 3.5 hours to be solved to the gap 1. A graph depicting the dependence of the runtime on N can be found in the full version of the paper. Sizes of the games range from 143 states and 2671 transitions to 1299 states and 34807 transitions.

Search Games

In search games that model intrusion, the defender patrols checkpoint zones (see Figure 1a, the zones are marked with box). The attacker aims to cross the graph, while not being captured by the defender. If the attacker crosses the graph unharmed, the defender receives a utility of -100. Whenever the attacker enters a node, she leaves a trace and the defender can later detect it. She can either wait for one move to conceal her presence (and clean up the trace), or move further.

We consider games with 2 checkpoint zones with varying sizes W (i.e. width of the graph) and 2 configurations of the defending forces – with one defender in each of the checkpoint zones (denoted 1-1), and 2 defenders in the first zone while just 1 defender being in the second one (denoted 2-1). The results are shown in Figure 1b (with 5 runs for each parameterization, the confidence intervals mark the standard error in our graphs). The largest game ($W = 5$ and 2 defenders in the first zone) has 4656 states and 121239 transitions and can be solved within 27 minutes. This case highlights that our algorithm can solve even large games. However, a much smaller game with the configuration 1-1 (964 states and 9633 transitions) is more challenging, since the coordination problem with just 1 defender in the first zone is harder, and is solved within 3.5 hours.

Patrolling Games

In patrolling games (Basilico, Gatti, and Amigoni 2009; Vorobeychik et al. 2014) the *patroller* patrols vertices of a graph by moving over the graph. The attacker decides the vertex she will attack and the time she will do so. The patroller does not know if an attack has started, however, he has a limited time (termed *attack time*, denoted $t_\times$) to reach the vertex under the attack. Otherwise, the vertex is successfully attacked and the patroller receives a negative reward associated to that vertex.

Following the setting in (Vorobeychik et al. 2014), we focus on graphs generated from Erdos-Renyi model (Newman 2010) with parameter $p = 0.25$ (denoted $ER(0.25)$) with attack times 3 and 4 and number of vertices $|\mathcal{V}|$ ranging from 7 to 15. Each instance with attack time $t_\times = 3$ was solved by our algorithm in less than 12 minutes (see Figure 1c). This result generally outperforms the computation times reported for tailored algorithm for solving discounted patrolling games (Vorobeychik et al. 2014). For attack time $t_\times = 4$, however, some number of instances failed to reach the precision $\mathrm{gap}(\hat{v}(b^0)) \leq 1$ within the time limit of 10 hours. For the most difficult setting, $|\mathcal{V}| = 13$, the algorithm reached desired precision in 60% of instances (see Figure 1d). For unsolved instances, mean $\mathrm{gap}(\hat{v}(b^0))$ after the cutoff after 10 hours is however reasonably small (also depicted in Figure 1d, see the solid line and right y-axes). The results include games with up to 856 states and 6409 transitions.

Since our algorithm is domain-independent, it can also solve variants of patrolling games with alarms (Basilico, Nittis, and Gatti 2016), including all types of imprecise signals (false positives, false negatives). The results for this setting can be found in the full version of the paper.

Conclusions

We focus on two-player zero-sum partially observable stochastic games (POSGs) with discounted rewards and one-sided observability where the second player has perfect information about the game. We propose the first approximate

algorithm that generalizes the ideas behind point-based algorithms designed for Partially Observable Markov Decision Processes (POMDPs) and transfers these techniques to POSGs. We provide theoretical guarantees as well as an experimental evaluation of our algorithm on three fundamentally different games.

Our work opens a completely new direction in research of POSGs and sequential decision making and allows to design new scalable algorithm for one-sided POSGs that can be applied in many real-world scenarios. While the current scalability of our algorithm is limited, it is the first step in a new direction of research. Many heuristics proven useful for POMDPs can be translated and evaluated in this new setting, and can further improve the scalability and applicability of our results.

Acknowledgments

This research was supported by the Czech Science Foundation (grant no. 15-23235S) and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/235/OHK3/3T/13.

References

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 57–64.

Basilico, N.; Nittis, G. D.; and Gatti, N. 2016. A Security Game Combining Patrolling and Alarm-Triggered Responses Under Spatial and Detection Uncertainties. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI)*, 397–403.

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* 51:829–866.

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

Ciesielski, K. 2007. On Stefan Banach and some of his results. *Banach Journal of Mathematical Analysis* 1(1):1–10.

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 *Proceedings of the 30th Conference on Artificial Intelligence (AAAI)*, 3966–3973.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *In Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2589–2595.

Hales, T. C. 2011. Historical overview of the Kepler conjecture. In *The Kepler Conjecture*. Springer. 65–82.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, 709–715.

Horak, K., and Bosansky, B. 2016. A Point-Based Approximate Algorithm for One-Sided Partially Observable Pursuit-Evasion Games. In *Proceedings of the Conference on Decision and Game Theory for Security*, 435–454.

Horak, K., and Bosansky, B. 2017. Dynamic Programming for One-Sided Partially Observable Pursuit-Evasion Games. In *Proceeding of the International Conference on Agents and Artificial Intelligence (ICAART) — to appear*.

Kiekintveld, C.; Jain, M.; Tsai, J.; Pita, J.; Ordóñez, F.; and Tambe, M. 2009. Computing optimal randomized resource allocations for massive security games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 689–696.

Madani, O.; Hanks, S.; and Condon, A. 1999. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of the 16th National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, 541–548.

Newman, M. 2010. *Networks: an introduction*. Oxford university press.

Nikaido, H. 1954. On von Neumann's minimax theorem. *Pacific J. Math.* 4(1):65–72.

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 *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 125–132.

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 *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 13–20.

Smith, T., and Simmons, R. 2004. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 520–527. AUAI Press.

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 *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 902–915.

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 *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 314–322.