Compact Representation of Value Function in Partially Observable Stochastic Games

Karel Horák[1] , **Branislav Bošanský**[1] , **Christopher Kiekintveld**[2] and **Charles Kamhoua**[3]
[1]Czech Technical University in Prague, FEE, Department of Computer Science
[2]The University of Texas at El Paso, Computer Science Department
[3]Army Research Laboratory, Network Security Branch
{horak, bosansky}@agents.fel.cvut.cz, cdkiekintveld@utep.edu, charles.a.kamhoua.civ@mail.mil

Abstract

Value methods for solving stochastic games with partial observability model the uncertainty of the players as a probability distribution over possible states, where the dimension of the belief space is the number of states. For many practical problems, there are exponentially many states which causes scalability problems. We propose an abstraction technique that addresses this curse of dimensionality by projecting the high-dimensional beliefs onto characteristic vectors of significantly lower dimension (e.g., marginal probabilities). Our main contributions are (1) a novel compact representation of the uncertainty in partially observable stochastic games and (2) a novel algorithm using this representation that is based on existing state-of-the-art algorithms for solving stochastic games with partial observability. Experimental evaluation confirms that the new algorithm using the compact representation dramatically increases scalability compared to the state of the art.

1 Introduction

Partially Observable Stochastic Games (POSGs) are a very general model of dynamic multi-agent interactions under uncertainty, and they can be used for modeling dynamic problems where players can react to other players based on limited, imperfect observations. Examples include patrolling [Basilico *et al.*, 2009; Vorobeychik *et al.*, 2014; Brazdil *et al.*, 2018] where a defender protects a set of targets against an attacker, pursuit-evasion games [Chung *et al.*, 2011] where a pursuer is trying to find and apprehend an evader. Security games including green security games [Fang *et al.*, 2015; 2016] and cybersecurity games [Nguyen *et al.*, 2017]) can also be generalized so that the defender can observe and react to attackers during the game.

Unfortunately, computing optimal strategies in POSGs is highly intractable from the algorithmic perspective, even in the two-player zero-sum setting. When both players have partial information about the environment, the players may need to reason not only about their belief over possible states, but also about the belief the opponent has over the possible states, *beliefs over beliefs*, and so on. Restricting to subclasses of POSGs where this issue of *nested beliefs* does not arise allows us to design and implement algorithms that are guaranteed to converge to optimal strategies [Horák *et al.*, 2017; Horák and Bošanský, 2019]. However, the scalability of the current algorithms even for this case is limited.

One of the fundamental problems is the complexity of representing and reasoning about uncertainty over a potentially very large state space. In these POSGs (as in single-agent Partially Observable Markov Decision Processes (POMDPs)), beliefs are probability distributions over the possible states. This is a well-known disadvantage of these models, since memory and computation time grow rapidly due to the *curse of dimensionality*. Taking a related approach to previous work on POMDPs (e.g., in [Roy *et al.*, 2005; Li *et al.*, 2010; Zhou *et al.*, 2010]), we address this problem by introducing a compact representation of the uncertainty in POSGs, and we develop a novel algorithm based on this representation that dramatically improves the scalability.

As a motivating domain, we consider a cybersecurity example where an attacker uses lateral movement actions to expand his control in a network without being detected. The defender tries to observe the attacker and take actions to protect the network by reconfiguring honeypots. A perfect-information version of this problem was proposed in [Kamdem *et al.*, 2017], however, we focus on a more realistic model where the defender has a limited information about the attacker. This version of the lateral movement game with uncertainty can be modeled as a one-sided POSG (OS-POSG) [Horák *et al.*, 2017]. Originally, the belief of the defender is defined over the possible subsets of resources that the attacker may currently control in the network. This representation scales exponentially in the size of the network so it is intractable for all but the smallest examples. We return to our motivating domain in Section 4 where key steps of our novel algorithm are discussed specifically for this domain.

Our main technical contribution is replacing the representation of beliefs over the exponential number of possible states using a compact *characteristic vector* that captures key information but reduces the dimensionality of the beliefs. Specifically, we propose using the marginal probability of each resource being infected as a characteristic vector instead of explicitly considering all possible subsets of infected resources. While this offers a path to scaling to much larger problems, it has consequences for the solution quality as well as the construction of the solution algorithm. Many components of state-of-the-art POSG solvers are based on manipulating the full belief distribution and we show how these can be redesigned to operate on more compact characteristic vectors. We formally define the fixed-point equation, show that the value function in the compact representation is still a convex function, and that solving a compact representation of the game yields a lower bound on the value of the original game. For our motivating domain, the novel algorithm operating on the compact representation scales orders of magnitude better with a negligible loss in quality (less than 1%) compared to the state of the art algorithm.

2 One-sided POSGs

A *one-sided partially observable stochastic game* [Horák *et al.*, 2017], or OS-POSG, is an imperfect-information two-player zero-sum infinite-horizon game with perfect recall represented by a tuple (S, A_1, A_2, O, T, R). The game is played for an infinite number of *stages*. At each stage, the game is in one of the states $s \in S$ and the players choose their actions $a_1 \in A_1$ and $a_2 \in A_2$ simultaneously. An initial state of the game is drawn from a probability distribution $b^0 \in \Delta(S)$ over states termed the *initial belief*. The one-sided nature of the game translates to the fact that while player 1 lacks detailed information about the course of the game, player 2 is able to observe the game perfectly (i.e., his only uncertainty is the action a_1 player 1 is about to take in the current stage).

The choice of actions determines the outcome of the current stage: player 1 gets an *observation* $o \in O$ and the game transitions to a state $s' \in S$ with probability $T(o, s' \mid s, a_1, a_2)$, where s is the current state. Furthermore, player 1 gets a reward $R(s, a_1, a_2)$ for this transition, and player 2 receives $-R(s, a_1, a_2)$. The rewards are discounted over time with discount factor $\gamma < 1$.

OS-POSGs can be solved by approximating the optimal value function $V^* : \Delta(S) \to \mathbb{R}$, mapping beliefs $b \in \Delta(S)$ of player 1 to his expected utility in belief b, using a pair of value functions $\underline{V}$ and $\overline{V}$ (lower and upper bound on V^*). The algorithm from [Horák *et al.*, 2017] refines these bounds by solving a sequence of *stage games* and is guaranteed to converge to an ϵ-approximation of the optimal solution. In each of these stage games, the algorithm finds the optimal strategies of the players in this stage (i.e., $\pi_1 \in \Delta(A_1)$ for player 1 and $\pi_2 : S \to \Delta(A_2)$ for player 2) while assuming that the current belief of player 1 is b and the play in the subsequent stages yields values represented by value functions $\underline{V}$ or $\overline{V}$, respectively. Solving the stage games also defines the fixed point equation for V^* w.r.t. the dynamic operator H,

$$V^*(b) = HV^*(b) = \min_{\pi_2} \max_{\pi_1} \Big(\mathbb{E}_{b,\pi_1,\pi_2}[R(\cdot)] + \qquad (1)$$

$$+ \gamma \sum_{a_1,o} \mathrm{Pr}_{b,\pi_1,\pi_2}[a_1, o] \cdot V^*(\tau(b, a_1, \pi_2, o)) \Big) ,$$

where $\tau(b, a_1, \pi_2, o)$ denotes the Bayesian update of the belief of player 1 when he played a_1 and observed o. Recall that observations of player 1 depend both on the states of the game and actions of player 2 who has complete information–hence the dependency of $\mathrm{Pr}_{b,\pi_1,\pi_2}[a_1, o]$ on b and π_2.

For piecewise-linear and convex $\underline{V}$ and $\overline{V}$, a stage game can be solved using linear programming. We show this linear program for $\underline{V}$ (represented as a point-wise maximum over a set Γ of linear functions $\alpha_i : \Delta(S) \to \mathbb{R}$).

$$\min\ V, \qquad \text{subject to:} \qquad (2a)$$

$$\sum_{a_2} \pi_2(s \wedge a_2) = b(s) \qquad \forall s \quad (2b)$$

$$V \geq \sum_{s,a_2} \pi_2(s \wedge a_2) R(s, a_1, a_2) + \gamma \sum_{o} V_{a_1 o} \qquad \forall a_1 \quad (2c)$$

$$b^{a_1 o}(s') = \sum_{s,a_2} \pi_2(s \wedge a_2) T(o, s' | s, a_1, a_2) \quad \forall a_1, o, s' \quad (2d)$$

$$V_{a_1 o} \geq \sum_{s'} \alpha_i(s') b^{a_1 o}(s') \qquad \forall a_1, o, \alpha_i \quad (2e)$$

$$\pi_2(s \wedge a_2) \geq 0 \qquad \forall s, a_2 \quad (2f)$$

In this linear program, player 2 seeks a strategy π_2 for the current stage of the game to minimize the utility V of player 1. Here $\pi_2(s \wedge a_2)$ stands for the joint probability (ensured by (2b) and (2f)) that the current state of the game is s and player 2 plays the action a_2. Constraint (2c) stands for player 1 choosing the best-responding action a_1. Constraint (2d) expresses the belief $b^{a_1 o}$ in the subsequent stage of the game when a_1 was played by player 1 and observation o was seen (multiplied by the probability of seeing that observation), and finally constraint (2e) represents the value of $\underline{V}$ in the belief $b^{a_1 o}$. Such a linear program can be then combined with point-based variants of the value-iteration algorithm, such as Heuristic Search Value Iteration (HSVI) [Smith and Simmons, 2004; Horák *et al.*, 2017].

A different approximation scheme is used for the upper bound $\overline{V}$ on V^*. Instead of using the point-wise maximum over a set of linear functions $\alpha_i \in \Gamma$, $\overline{V}$ is expressed using a set $\Upsilon = \{(b^{(i)}, y^{(i)}) \mid 1 \leq i \leq |\Upsilon|\}$ of points (where $b^{(i)}$ is the coordinate of a point in the belief space, and $y^{(i)}$ is its associated value). The lower convex hull of this set of points is then formed to obtain the value of $\overline{V}(b)$,

$$\overline{V}(b) = \min_{\lambda \in \mathbb{R}^{|\Upsilon|}_{\geq 0}} \left\{ \sum_{1 \leq i \leq |\Upsilon|} \lambda_i y^{(i)} \mid \mathbf{1}^T \lambda = 1, \sum_{1 \leq i \leq |\Upsilon|} \lambda_i b^{(i)} = b \right\} . \qquad (3)$$

Here, λ stands for the coefficients of a convex combination of points in Υ (requiring that the convex combination of the coordinates $b^{(i)}$ of the points matches the belief b). Constraint (2e) can then be adapted to use this representation of $\overline{V}$ as detailed in [Horák and Bošanský, 2016].

3 Compact Representation of V^*

The dimension of the value function V^* is the number of states, which is exponential in the size of the network for our motivating domain. We propose an abstraction scheme called *summarized abstraction* to decrease the dimensionality of the problem by creating a simplified representation of the beliefs over the state space.

We associate each belief $b \in \Delta(S)$ in the game with its *characteristic vector* $\chi^{(b)} = \mathbf{A} \cdot b$ (for some fixed matrix

$\mathbf{A} \in \mathbb{R}^{k \times |S|}$ where $k \ll |S|$) and we define an (approximate) value function $\tilde{V}^* : \mathbb{R}^k \to \mathbb{R}$ over characteristic vectors.

The main goal is to adapt algorithms based on value iteration to operate over the more compact space $\mathbb{R}^k$ instead of the original belief space $\Delta(S)$. First, we adapt the fixed point equation (1) for OS-POSGs to work with compact $\tilde{V}^*$ (instead of original V^*). We let the player 2 choose *any* belief that is consistent with the current characteristic vector χ (by adding an extra minimization term), and we replace the value of belief b, $V^*(b)$, by the value of its characterization $\tilde{V}^*(\mathbf{A} \cdot b)$. We also denote this compound function $\tilde{V}^* \circ \mathbf{A}$.

$$\tilde{V}^*(\chi) = \tilde{H}\tilde{V}^*(\chi) = \min_{b|\mathbf{A}b=\chi} \min_{\pi_2} \max_{\pi_1} \Big(\mathbb{E}_{b,\pi_1,\pi_2}[R] + \tag{4}$$
$$+ \gamma \sum_{a_1,o} \mathrm{Pr}_{b,\pi_1,\pi_2}[a_1, o] \cdot \tilde{V}^*(\mathbf{A} \cdot \tau(b, a_1, \pi_2, o)) \Big)$$

Next, we show that the value function $\tilde{V}^*$ satisfying the fixed point equation (4) (and obtained as a limit of applying operator $\tilde{H}$) provides a valid lower bound on the solution of the original game representation over the belief space.

Theorem 1. *$\tilde{V}^*(\chi^{(b)}) \leq V^*(b)$ for every $b \in \Delta(S)$.*

Proof sketch. Let $\tilde{V}_0$ be arbitrary and $V_0(b) = \tilde{V}_0(\chi^{(b)})$. Construct sequences $\{\tilde{V}_t\}_{t=0}^{\infty}$ and $\{V_t\}_{t=0}^{\infty}$ such that $\tilde{V}_{t+1} = \tilde{H}\tilde{V}_t$ and $V_{t+1} = HV_t$. Assume $\tilde{V}_t(\chi^{(b)}) \leq V_t(b)$ for every b (which holds for $t = 0$). Now $H(\tilde{V}_t \circ \mathbf{A}) \leq HV_t$. The extra minimization over beliefs b in $\tilde{H}\tilde{V}_t$ can only decrease the utility and hence $\tilde{V}_{t+1}(\chi^{(b)}) = \tilde{H}\tilde{V}_t(\chi^{(b)}) \leq H(\tilde{V}_t \circ \mathbf{A})(b) \leq HV_t(b) = V_{t+1}(b)$. This extends to the limits $\tilde{V}^*$ and V^*. □

Note that the equation (4) can also be solved using linear programming. Consider a lower bound $\tilde{V}_{\mathrm{LB}}$ on $\tilde{V}^*$ formed as a point-wise maximum over linear functions $\alpha_i(\chi) = (\mathbf{a}^{(i)})^T\chi + z^{(i)}$. We modify the linear program (2) by considering that the belief b is a variable (constrained by χ),

$$\text{constraints (2b), (2c), (2d), (2f)} \tag{5a}$$
$$\sum_s b(s) = 1 \tag{5b}$$
$$\mathbf{A} \cdot b = \chi \tag{5c}$$
$$b(s) \geq 0 \qquad \forall s\,, \tag{5d}$$

and we replace the constraint (2e) to account for different representation of $\tilde{V}_{\mathrm{LB}}$ by

$$\chi^{a_1 o} = \mathbf{A} \cdot b^{a_1 o} \qquad \forall a_1, o \tag{5e}$$
$$q^{a_1 o} = \mathbf{1}^T \cdot b^{a_1 o} \qquad \forall a_1, o \tag{5f}$$
$$V_{a_1 o} \geq (\mathbf{a}^{(i)})^T \cdot \chi^{a_1 o} + z^{(i)} \cdot q^{a_1 o} \qquad \forall a_1, o, \alpha_i\,, \tag{5g}$$

where $q^{a_1 o} = \mathbf{1}^T \cdot b^{a_1 o}$ is the probability that o is generated when player 1 uses action a_1.

Observe that the only constraints with non-zero right-hand side in this new linear program are constraints (5b) and (5c). This is critical in the following proof that $\tilde{V}^*$ is convex.

```
Algorithm 1: HSVI algorithm for OS-POSGs when summarized abstraction is used.
1  Initialize $\tilde{V}_{\mathrm{LB}}$ and $\tilde{V}_{\mathrm{UB}}$ to lower and upper bound on $\tilde{V}^*$
2  while $\tilde{V}_{\mathrm{UB}}(\chi^0) - \tilde{V}_{\mathrm{LB}}(\chi^0) > \epsilon$ do
3      Explore($\chi^0, \epsilon, 0$)
4  procedure Explore($\chi, \epsilon, t$)
5      $(b, \pi_2) \leftarrow$ optimal belief and strategy of player 2 in
         $\tilde{H}\tilde{V}_{\mathrm{LB}}(\chi)$
6      $(a_1, o) \leftarrow$ select according to heuristic
7      $\chi' \leftarrow \tau(\chi, a_1, \pi_2, o)$
8      Update $\Gamma$ and $\Upsilon$ based on the solutions of $\tilde{H}\tilde{V}_{\mathrm{LB}}(\chi)$
         and $\tilde{H}\tilde{V}_{\mathrm{UB}}(\chi)$
9      if $\tilde{V}_{\mathrm{UB}}(\chi') - \tilde{V}_{\mathrm{LB}}(\chi') > \epsilon\gamma^{-t}$ then
10         Explore($\chi', \epsilon, t + 1$)
11         Update $\Gamma$ and $\Upsilon$ based on the solutions of
             $\tilde{H}\tilde{V}_{\mathrm{LB}}(\chi)$ and $\tilde{H}\tilde{V}_{\mathrm{UB}}(\chi)$
```

Theorem 2. *Value function $\tilde{V}^*$ is convex.*

Proof. Consider a dual formulation of the linear program (2) updated according to equations (5). Since the only non-zero right-hand side terms in the primal are 1 and χ, the objective of the dual formulation is $o(\chi) = \chi^T \cdot \mathbf{a} + z$. Moreover, this is the only place where the characteristic vector χ occurs. Hence the polytope of the feasible solutions of the dual problem is the same for every characteristic vector $\chi \in \mathbb{R}^k$ and $o(\chi)$ (after fixing variables $\mathbf{a}$ and z) forms a lower bound on the objective value of the solution for *arbitrary* χ. Since we maximize over all possible $o(\chi)$ in the dual, the objective value of the linear program (and also $H\tilde{V}_{\mathrm{LB}}$) is convex in the parameter χ.

Now, starting with an arbitrary convex (e.g., linear) $\tilde{V}^0_{\mathrm{LB}}$, the sequence of functions $\{\tilde{V}^t_{\mathrm{LB}}\}_{t=0}^{\infty}$, where $\tilde{V}^{t+1}_{\mathrm{LB}} = \tilde{H}\tilde{V}^t_{\mathrm{LB}}$, is formed only by convex functions. Therefore, the fixed point $\tilde{V}^*$ is also a convex function. □

3.1 HSVI Algorithm for Compact POSGs

The Algorithm 1 we propose for solving abstracted games is a modified version of the original heuristic search value iteration algorithm (HSVI) for solving unabstracted OS-POSGs [Horák *et al.*, 2017]. The key difference is that we use the value functions $\tilde{V}_{\mathrm{LB}}$ and $\tilde{V}_{\mathrm{UB}}$ (instead of $\underline{V}$ and $\overline{V}$) and we have modified all parts of the algorithm to use the abstracted representation of the beliefs.

First, we initialize bounds $\tilde{V}_{\mathrm{LB}}$ and $\tilde{V}_{\mathrm{UB}}$ (line 1) to valid piecewise linear and convex lower and upper bounds on $\tilde{V}^*$ (using sets Γ of linear functions $\alpha_i = (\mathbf{a}^{(i)})^T\chi + z^{(i)}$ and Υ of points $(\chi^{(i)}, y^{(i)})$). Then, we perform a sequence of trials (lines 2–3) from the initial characteristic vector $\chi^0 = \mathbf{A}b^0$ until the desired precision $\epsilon > 0$ is reached.

In each of the trials, we first compute the optimal *optimistic* strategy of player 2, which in this case is the selection of the belief b and strategy π_2 (line 5). Next, we choose the action a_1 of player 1 and the observation o (lines 6–7) so that the excess approximation error $\tilde{V}_{\mathrm{UB}}(\chi') - \tilde{V}_{\mathrm{LB}}(\chi') - \epsilon\gamma^{-t}$ in the subsequent stage (where the belief is described by a characteristic vector $\chi' = \tau(\chi, a_1, \pi_2, o))$ is maximized. If this excess approximation error is positive, we recurse to the characteristic vector χ' (line 10).

Before and after the recursion we update the bounds $\tilde{V}_{\mathrm{LB}}(\chi)$ and $\tilde{V}_{\mathrm{UB}}(\chi)$ using the dynamic operator $\tilde{H}$ (lines 8 and 11). The update of $\tilde{V}_{\mathrm{UB}}$ is straightforward and a new point $(\chi, \tilde{H}\tilde{V}_{\mathrm{UB}}(\chi))$ is added to Υ. To obtain a new linear function to add to the set Γ, we use the objective function $o(\chi) = \chi^T \cdot \mathbf{a} + z$ (after fixing variables $\mathbf{a}$ and z) of the dual linear program to (5) that forms a lower bound on $\tilde{V}^*$.

4 The Lateral Movement POSG

We now specify the approach for a cybersecurity problem of detecting and mitigating lateral movement of an attacker in a network. The game is played on a directed acyclic graph $G = (V, E)$, where the vertices $V = \{v_1, \ldots, v_{|V|}\}$ are sorted in topological order. The goal of the attacker is to reach vertex $v_{|V|}$ from the initial source of the infection v_1 by traversing the edges of the graph, while minimizing the cost to do so.

Initially, the attacker only controls the vertex v_1, i.e., the initial *infection* is $I_0 = \{v_1\}$. In every stage of the game, the attacker chooses a directed path $P = \{P(i)\}_{i=1}^{k}$ (where k is the length of the path) from any of the infected vertices to the target vertex $v_{|V|}$. Unless the defender takes countermeasures, the attacker infects all of the vertices on the path, including the target vertex $v_{|V|}$, which ends the game. The attacker pays the total cost of traversing the edges on the path,

$$c_{-,P} = \sum_{P(i)\in P} C(P(i)) \,, \tag{6}$$

where $C(P(i))$ is a cost associated with taking edge $P(i)$.

The defender tries to discover the attacker and increase his costs by deploying a honeypot in the network. The honeypot is deployed on an edge (denote this edge h) of the graph, and is able to detect if the attacker traverses that specific edge. Furthermore, it also increases the cost for using the edge h to $\overline{C}(h)$. If the attacker observes that he has traversed a honeypot, he may decide to change his plan and therefore does not execute the rest of his originally intended path P. The cost of playing a path P against a honeypot placement h is therefore

$$c_{h,P} = \sum_{P(i)\in P_{\leq h}} C(P(i)) + \mathbf{1}_{h\in P} \cdot [\overline{C}(h) - C(h)] \tag{7}$$

where $P_{\leq h}$ is the prefix of the path P until the interaction with the honeypot edge h,

$$P_{\leq h} = \begin{cases} P & h \notin P \\ \{P(i)\}_{i=1}^{\bar{i}} \text{ where } P(\bar{i}) = h & \text{otherwise} \end{cases} . \tag{8}$$

Since the attacker does not necessarily continue to execute his selected path P (because the defender can reconfigure the position of the honeypot), the new infection $I_{h,P}$ (i.e., state of the game) becomes

$$I_{h,P} = I \cup \{v \mid (u, v) \in P_{\leq h}\} \ . \tag{9}$$

This problem can be formalized as a OS-POSG where the states are possible infections ($S = 2^V$), defender actions are honeypot allocations ($A_1 = E$), and actions A_2 of the attacker are paths in G reaching $v_{|V|}$. Observations denote whether the defender detected the attacker (i.e., $h \in P$) or the attacker has reached the target $v_{|V|}$ while avoiding the detection, $O = \{\mathrm{det}, \neg\mathrm{det}\}$[1]. Transitions follow the equation (9) and observation det is generated iff the honeypot edge h has been traversed. The reward of the defender is the negative value of the cost of the attacker ($R(h, P) = -c_{h,P}$), the discount factor of the game is $\gamma = 1$ [2], and the initial belief b^0 of the game satisfies $b^0(I_0) = 1$.

4.1 Characteristic Vectors

The number of states in the game is exponential in the number of vertices of the graph, $|S| = 2^{|V|-2} + 1$ (we consider that v_1 is always infected and we treat all states where $v_{|V|}$ is infected as a single terminal state of the game). We propose using the marginal probabilities of a vertex being infected as characteristic vectors $\chi \in \mathbb{R}^{|V|}$, i.e.

$$\chi_i^{(b)} = \sum_{I\in S \mid v_i \in I} b(I) \,, \tag{10}$$

where I corresponds to some subset of possibly infected vertices of the graph and $b(I)$ denotes the original belief over this subset of vertices being infected. Note that the linear projection introduced in equation (10) defines a projection matrix $\mathbf{A} \in \mathbb{R}^{|V|\times|S|}$ from Section 3.

4.2 Value Function Representation

The algorithm from Section 3.1 approximates $\tilde{V}^*$ using a pair of value functions, the lower bound $\tilde{V}_{\mathrm{LB}}$ and the upper bound $\tilde{V}_{\mathrm{UB}}$. While we have discussed representing the lower bound using a set Γ of linear functions $\alpha_i(\chi) = (\mathbf{a}^{(i)})^T\chi + z^{(i)}$, representing the upper bound is more challenging. In the original algorithm, the value function $\overline{V}$ is defined (see equation (3)) over the probability simplex $\Delta(S)$. In that case, it suffices to consider $|S|$ points in Υ to define $\overline{V}$ for every belief. In contrast, the space of characteristic vectors here (i.e., marginal probabilities) is formed by a hypercube $[0,1]^{|V|}$ with $2^{|V|}$ vertices, which makes the straightforward point representation (using equation (3)) impractical.

However, we can leverage the fact that in this domain infecting an additional node can only decrease the cost to the target (and hence $\tilde{V}^*$ is decreasing). Consider the dual formulation of the optimization problem (3). In this formulation, the projection of χ to the lower convex hull of a set of points is represented by the *optimal* linear function $\mathbf{a}^T\chi + z$ defining a facet of the convex hull (see Figure 1). Since $\tilde{V}^*$ is decreasing in χ, we can also enforce that $\mathbf{a}^T\chi + z$ is decreasing in χ (i.e., add the constraint $\mathbf{a} \leq 0$ to the dual formulation).

[1] Since we assume a perfect-recall game, the defender knows the current allocation of the honeypot and is able to infer the edge where the attacker has been detected upon receiving det observation.

[2] While the original HSVI algorithm for OS-POSGs has been defined and proven for problems with $\gamma < 1$, the convergence properties translate even to the undiscounted case in this case since the game is essentially finite (in a finite number of steps, *all* vertices, including $v_{|V|}$, get infected and the game ends).

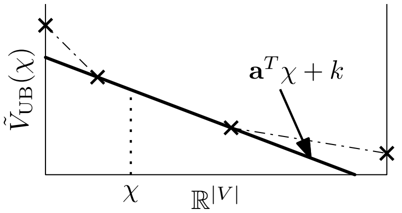

Figure 1: Dual interpretation of the projection on the convex hull.

This additional constraint translates to changing the equality $\sum_{1\leq i\leq|\Upsilon|}\lambda_i\chi^{(i)} = \chi$ in the primal problem to an inequality.

$$\tilde{V}_{\mathrm{UB}}(b) = \min_{\lambda\in\mathbb{R}^{|\Upsilon|}_{\geq 0}}\left\{\sum_{1\leq i\leq|\Upsilon|}\lambda_i y^{(i)} \mid \mathbf{1}^T\lambda = 1, \sum_{1\leq i\leq|\Upsilon|}\lambda_i\chi^{(i)} \leq \chi\right\} \tag{11}$$

Now it is sufficient that the set Υ contains just one point $(\chi^{(i)}, y^{(i)})$ where $\chi^{(i)} = \mathbf{0}^{|V|}$ (instead of $2^{|V|}$ points) to make the constraint $\sum_{1\leq i\leq|\Upsilon|}\lambda_i\chi^{(i)} \leq \chi$ satisfiable.

It is possible to adapt the constraint (5g) to use the representation from (11), similarly to the original (unabstracted) OS-POSGs [Horák and Bošanský, 2016], and obtain a linear program to solve the stage games w.r.t. the upper bound.

4.3 Using Marginalized Strategies in Stage Games

The linear program formed by modifications from equations (5) still requires solving the stage game for the original, unabstracted problem. In this section, we show that it is possible to avoid expressing the belief b explicitly, and to compute the stage game directly using the characteristic vectors and marginalized strategies of the attacker.

First, we present the representation of the stage-game strategies of the attacker. Instead of representing joint probabilities $\pi_2(I \wedge P)$ of choosing path P in state I, we only model the marginalized probability $\tilde{\pi}_2(P)$ of choosing path P aggregated over all states $I \in S$. Furthermore, we allow the attacker to choose the probability $\xi(P \wedge v_i)$ that vertex v_i is infected while he opts to follow path P.

$$\sum_P \tilde{\pi}_2(P) = 1 \tag{12a}$$

$$0 \leq \xi(P \wedge v_i) \leq \tilde{\pi}_2(P) \qquad \forall P, v_i \tag{12b}$$

$$\tilde{\pi}_2(P) \geq 0 \qquad \forall P \tag{12c}$$

To ensure that the strategy represented by variables $\tilde{\pi}_2$ and ξ is feasible it must be consistent with the characteristic vector χ, where χ_i is the probability that the vertex v_i is infected at the beginning of the stage. By marginalizing the variables $\xi(\cdot)$ representing the joint probabilities we get

$$\sum_P \xi(P \wedge v_i) = \chi_i \qquad \forall v_i\ . \tag{12d}$$

Furthermore, the path P must start in an already infected vertex (denoted as $\mathrm{Pre}(P)$), i.e., the conditional probability $\Pr[\mathrm{Pre}(P) \in I \mid P]$ of $\mathrm{Pre}(P)$ being infected when path P is chosen has to be 1. Now, since $\xi(P \wedge v)$ is the joint probability, $\xi(P \wedge v) = \Pr[\mathrm{Pre}(P) \in I \mid P] \cdot \tilde{\pi}_2(P)$,

$$\xi(P \wedge v) = \tilde{\pi}_2(P) \qquad \forall P, v = \mathrm{Pre}(P)\ . \tag{12e}$$

This representation of attacker strategies is sufficient to express the expected immediate reward of the strategy $\tilde{\pi}_2$, hence the constraint (2c) can be changed to use the marginalized strategies,

$$V \geq \sum_P \tilde{\pi}_2(P)c_{h,P} + \sum_o V_{ho} \qquad \forall h \in E\ . \tag{12f}$$

Importantly, we can also skip the computation of the belief b^{ho} and compute the characteristic vector formed by the marginals χ^{ho} directly from the variables $\tilde{\pi}_2$ and ξ. We now present the equation to compute the updated marginal $\chi^{h,\mathrm{det}}$ given that the attacker has been detected while traversing the honeypot edge h.

$$\chi_i^{h,\mathrm{det}} = \sum_{P|h\in P\wedge P_{\leq(\cdot,v_i)}\subseteq P_{\leq h}} \tilde{\pi}_2(P) \qquad + \sum_{P|h\in P\wedge P_{\leq(\cdot,v_i)}\nsubseteq P_{\leq h}} \xi(P \wedge v_i) \tag{12g}$$

The first sum stands for the probability that the attacker is detected while traversing edge h, but he infected v_i beforehand. The second sum represents the probability that the attacker used edge h as well, but this time he has not infected v_i using path P, however, the vertex v_i has already been infected before starting to execute path P.

Analogously, we can obtain the probability $q^{h,\mathrm{det}}$ that the attacker got detected while traversing edge h as

$$q^{h,\mathrm{det}} = \sum_{P|h\in P} \tilde{\pi}_2(P)\ . \tag{12h}$$

We need not consider the subsequent stages where the attacker has not been detected (i.e., $\neg\mathrm{det}$ observation has been generated) or the honeypot edge h reaches the target vertex $v_{|V|}$. In both of these cases, the target vertex has been reached and thus the value of the subsequent stage is zero.

4.4 Initializing Bounds

We now describe how we initialize bounds $\tilde{V}_{\mathrm{LB}}$ and $\tilde{V}_{\mathrm{UB}}$ (line 1 of Algorithm 1). Denote by $C^*(v_i)$ and $\overline{C}^*(v_i)$ the costs of the shortest (i.e., cheapest) paths from v_i to the target $v_{|V|}$ when costs C and $\overline{C}$, respectively, are used for all edges.

We initialize the lower bound $\tilde{V}_{\mathrm{LB}}$ using two linear functions $\alpha_1(\chi) = 0$ and $\alpha_2(\chi) = (\mathbf{a}^{(2)})^T\chi + z^{(i)}$ such that $z^{(i)} = C^*(v_1)$ and $\mathbf{a}_i^{(2)} = \min\{C^*(v_i) - C^*(v_1), 0\}$.

To initialize the upper bound $\tilde{V}_{\mathrm{UB}}$, we consider that exactly one vertex v_i is infected and we consider the most expensive path $\overline{C}^*(v_i)$ from v_i to the target $v_{|V|}$. We use the set $\Upsilon = \{(\chi^{(j)}, y^{(j)}) \mid 1 \leq j \leq |V|\}$ of points to initialize $\tilde{V}_{\mathrm{UB}}$ where

$$\chi_i^{(j)} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad y^{(j)} = \overline{C}^*(v_j)\ . \tag{13}$$

5 Experimental Results

In this section we experimentally evaluate the scalability and properties of our proposed abstraction technique based on the model from Section 4. Unless otherwise stated, we consider directed acyclic graphs generated from the Erdős-Rényi model with probability $p = 0.5$ that each of the possible edges is included. Furthermore, we make sure to include

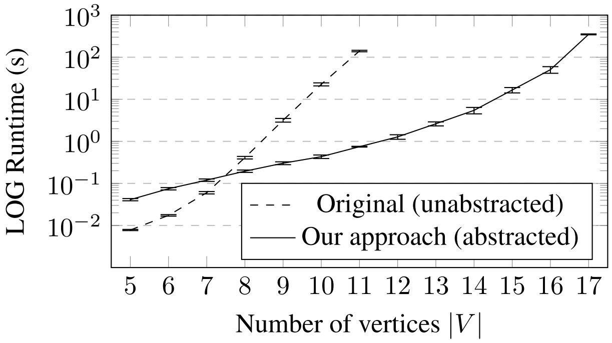

Figure 2: Comparison of the runtime of the original (unabstracted) approach with the proposed one (using summarized abstraction). Confidence intervals mark standard error.

edges (v_i, v_{i+1}) for $1 \leq i \leq |V| - 1$. We set the costs $C(v_i, v_j) = j - i$ for every edge $(v_i, v_j) \in E$ when the honeypot is not deployed to (v_i, v_j). We do, however, apply a significant penalty for traversing a honeypot edge where $\overline{C}(v_i, v_j) = j(j - i)$. This setting is used to model layered networks typical of critical infrastructure [Kuipers and Fabro, 2006]. The attacker can either proceed to the subsequent layer (using edge (v_i, v_{i+1})), or use a shortcut (if available). The costs $\overline{C}$ reflect the fact that the closer the attacker is to the target, the more secure the network is (i.e., he has to use a more expensive exploit to proceed).

All of the experiments were run on a machine with an Intel i7-8700K and 32GB of RAM. We used CPLEX 12.7.1 to solve the linear programs used in the algorithms.

5.1 Comparison with the Original Approach

We used a set of randomly generated graphs (varying the number of vertices) and we attempted to solve these instances using both the original (unabstracted) approach and the algorithm we present in this paper. The parameters used for the original algorithm follow the parameters proposed in [Horák *et al.*, 2017]. We modified the initialization of the bounds to make it valid for the undiscounted problem in question. The target precision $\epsilon = 0.1$ was used for both algorithms.

Figure 2 shows the runtimes of the original algorithm applied to the unabstracted game and our proposed approach. For small instances, the original approach is competitive and outperforms our proposed approach. For larger instances, however, the compact representation used in our approach turns out to be significantly better. Moreover, the original approach was unable to solve 50% of the 11-vertex instances due to memory requirements. In contrast, our approach based on marginal probabilities did not exceed 1GB of memory even when applied to the most challenging 17-vertex instances.

5.2 Abstraction Quality

We now focus on the abstraction quality since the summarized abstraction may lose information needed to derive the optimal behavior. However, we show that with properly designed characteristic vectors the quality loss can be minimal.

$\lvert V\rvert$	5	6	7	8	9	10
$\frac{\overline{V}(b^0)-\tilde{V}_{\mathrm{LB}}(\chi^0)}{\tilde{V}_{\mathrm{LB}}(\chi^0)}$ (in %)	0.8	0.9	1.0	0.7	0.7	0.5

Table 1: Empirical bound on the relative distance from the equilibrium of the unabstracted game based on 100 instances.

Due to the insufficient scalability of the original unabstracted approach, we can do the comparison only for graphs with $5 \leq |V| \leq 10$ vertices where the original approach solved all of the instances. In Table 1, we present the empirical bound on the relative distance from the equilibrium of the unabstracted bound. We compare the upper bound $\overline{V}(b^0)$ computed by the original approach for the unabstracted game with the lower bound $\tilde{V}_{\mathrm{LB}}(\chi^0)$ on the quality of the strategy when the abstraction is used. We depict the maximum relative error based on the 100 randomly generated instances for each size of a graph. In all of the cases, the empirical upper bound on the relative error (i.e., the worst-case possible quality loss due to abstraction) is below 1.0%. Note that in all of the instances the quality of the strategy found by our novel algorithm is within the bounds of the original approach (i.e., the empirical bounds are likely to be overestimated).

6 Conclusions

We focus on solving partially observable stochastic games (POSGs) and the representation of partial information in these games. In the existing algorithms, the dimension of the belief space is equal to the number of possible states. This fact limits the scalability since both required memory and computation time grow rapidly. We introduce a novel abstraction method and an algorithm relying on a compact representation of the uncertainty in these POSGs. Our methodology is domain-independent and we demonstrate it on a motivating example from cybersecurity where the defender protects a computer network against an attacker who uses lateral movement to reach the desired target. Experimental results show that our novel algorithm scales several orders of magnitude better compared to the existing state of the art with only a negligible loss in quality (less than 1%). Our paper demonstrates practical aspects of algorithms for solving subclasses of POSGs and opens the possibility of using similar compact representation for other domains.

Acknowledgments

This research was supported by the Czech Science Foundation (grant no. 19-24384Y) and by 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 *et al.*, 2009] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *8th International Conference on Autonomous Agents and Multiagent Systems*, pages 57–64, 2009.

[Brazdil *et al.*, 2018] Tomas Brazdil, Antonin Kucera, and Vojtech Rehak. Solving Patrolling Problems in the Internet Environment. In *27th International Joint Conference on Artificial Intelligence*, 2018.

[Chung *et al.*, 2011] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299–316, 2011.

[Fang *et al.*, 2015] Fei Fang, Peter Stone, and Milind Tambe. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *24th International Joint Conference on Artificial Intelligence*, 2015.

[Fang *et al.*, 2016] Fei Fang, Thanh Hong Nguyen, Rob Pickles, Wai Y. Lam, Gopalasamy R. Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. Deploying PAWS: Field optimization of the protection assistant for wildlife security. In *30th AAAI Conference on Artificial Intelligence*, pages 3966–3973, 2016.

[Horák and Bošanský, 2016] Karel Horák and Branislav Bošanský. A Point-Based Approximate Algorithm for One-Sided Partially Observable Pursuit-Evasion Games. In *International Conference on Decision and Game Theory for Security*, 2016.

[Horák and Bošanský, 2019] Karel Horák and Branislav Bošanský. Solving Partially Observable Stochastic Games with Public Observations. In *33rd AAAI Conference on Artificial Intelligence*, 2019.

[Horák *et al.*, 2017] Karel Horák, Branislav Bošanský, and Michal Pěchouček. Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games. In *31st AAAI Conference on Artificial Intelligence*, 2017.

[Kamdem *et al.*, 2017] Gael Kamdem, Charles Kamhoua, Yue Lu, Sachin Shetty, and Laurent Njilla. A markov game theoritic approach for power grid security. In *37th International Conference on Distributed Computing Systems Workshops*, pages 139–144. IEEE, 2017.

[Kuipers and Fabro, 2006] David Kuipers and Mark Fabro. *Control systems cyber security: Defense in depth strategies*. United States. Department of Energy, 2006.

[Li *et al.*, 2010] Xin Li, William K Cheung, and Jiming Liu. Improving pomdp tractability via belief compression and clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(1):125–136, 2010.

[Nguyen *et al.*, 2017] Thanh H. Nguyen, Michael P. Wellman, and Satinder Singh. A Stackelberg Game Model for Botnet Data Exfiltration. In *International Conference on Decision and Game Theory for Security*, 2017.

[Roy *et al.*, 2005] Nicholas Roy, Geoffrey Gordon, and Sebastian Thrun. Finding approximate pomdp solutions through belief compression. *Journal of artificial intelligence research*, 23:1–40, 2005.

[Smith and Simmons, 2004] Trey Smith and Reid Simmons. Heuristic Search Value Iteration for POMDPs. In *20th conference on Uncertainty in artificial intelligence*, 2004.

[Vorobeychik *et al.*, 2014] Yevgeniy Vorobeychik, Bo An, Milind Tambe, and Satinder P. Singh. Computing Solutions in Infinite-Horizon Discounted Adversarial Patrolling Games. In *24th International Conference on Automated Planning and Scheduling*, 2014.

[Zhou *et al.*, 2010] Enlu Zhou, Michael C Fu, and Steven I Marcus. Solving continuous-state pomdps via density projection. *IEEE Transactions on Automatic Control*, 55(5):1101–1116, 2010.