Dynamic Programming for One-sided Partially Observable Pursuit-evasion Games

Karel Horák and Branislav Bošanský
Department of Computer Science, Faculty of Electrical Engineering,
Czech Technical University in Prague, Prague, Czech Republic

Keywords: Pursuit-evasion Games, One-sided Partial Observability, Infinite Horizon, Value Iteration, Concurrent Moves.

Abstract: Pursuit-evasion scenarios appear widely in robotics, security domains, and many other real-world situations. We focus on two-player pursuit-evasion games with concurrent moves, infinite horizon, and discounted rewards. We assume that the players have partial observability, however, the evader has an advantage of knowing the current position of pursuer's units. This setting is particularly interesting for security domains where a robust strategy, maximizing the utility in the worst-case scenario, is often desirable. We provide, to the best of our knowledge, the first algorithm that provably converges to the value of a partially observable pursuit-evasion game with infinite horizon. Our algorithm extends well-known value iteration algorithm by exploiting that (1) value functions of our game depend only on the position of the pursuer and the belief he has about the position of the evader, and (2) that these functions are piecewise linear and convex in the belief space.

1 INTRODUCTION

Pursuit-evasion games (PEGs) appear in many scenarios in robotics and security domains (Vidal et al., 2002; Chung et al., 2011). A team of centrally controlled pursuing units (the *pursuer*) aims to locate and capture the *evader*, while the evader aims for the opposite. We study these games and assume their discrete-time variant played on a finite graph. We assume that units of both players move simultaneously, the horizon of the game is infinite, rewards are discounted over time with discount factor $\gamma \in [0,1)$, and the players have only a partial information about the current state. Formally, such a game belongs to zero-sum partially observable stochastic games (POSGs).

We aim for finding robust strategies of the pursuer against the worst-case evader. Specifically, we assume that the evader knows the positions of pursuer's units and her only uncertainty is the strategy of the pursuer and the move he will perform next. Although in reality such perfectly informed adversary is rarely met, the pursuer usually does not know what information is being revealed to the evader. Hence, in order to derive robust strategies (i.e. maximizing pursuer's reward against *any* type of the evader), it is natural to use such a perfectly informed adversary.

We design the first algorithm that provably converges to the value of such one-sided partially observable pursuit-evasion games. Moreover, as the value converges, strategies of the players converge to the optimal strategies as well. This contrasts with existing approaches in robotics and security, where heuristic solutions without any optimality guarantees are used (Vidal et al., 2002; Chung et al., 2011).

Our algorithm extends the well-known value iteration algorithms for concurrent-moves stochastic games (Shapley, 1953) and partially observable Markov decision processes (POMDPs) (Smallwood and Sondik, 1973; Monahan, 1982; Pineau et al., 2003; Smith and Simmons, 2012). We show that, similarly to POMDPs, one-sided pursuit-evasion games allow us to define compactly represented value functions and propose a dynamic programming approach to approximate them in an iterative manner. Specifically we show that the value functions (1) depend only on the position of the pursuer's units and his belief about the possible position of the evader, but *not* on the history of moves, (2) these functions are piecewise linear and convex and thus we can represent them as a set of α-vectors (Section 2.1), and (3) we can design a dynamic-programming operator with provable convergence to optimal value functions (Section 3).

Our results for one-sided partially observable pursuit-evasion games have similar implications as those derived for POMDPs. Our paper is thus the first step in a whole line of research. The importance of

HorÃąk K. and BoÅąanskÃ¡ B.
Dynamic Programming for One-sided Partially Observable Pursuit-evasion Games.
DOI: 10.5220/0006190605030510
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 503-510
ISBN: 978-989-758-220-2

the results is highlighted in the derivation of the full-backup value iteration algorithm. Moreover, due to the close similarity with POMDP models in the structure of the solution, efficient point-based versions of the algorithm should be applicable as well.

Due to the space constraints, proofs of some results can be found in the full version of the paper.

1.1 Related Work

A similar model with one-sided partial observability where one of the players has a perfect information was presented in (McEneaney, 2004). This player is assumed to know the action the opponent will play at the current stage. Such a game is essentially turn-based and only pure strategies are thus thought of.

Disregarding randomization has severe limitations. In many cases, if the evader knows the action of the pursuer before she has to decide herself, she can use this additional information to avoid getting caught (simply by avoiding the vertices the pursuer is about to move to next forever). Randomized strategies thus better correspond to real-world problems occurring in real-time. Using them, however, presents additional challenges that we solve in this paper.

Finite horizon POSGs can be also solved by converting a game to the matrix form by enumerating all pure strategies of the players. In (Hansen et al., 2004), the pure strategies are constructed in an incremental way using dynamic programming while prunning those that are dominated. Although this improves on the naïve enumeration approach, the number of strategies is still exponential in the horizon in the worst case and so is their size, which makes the algorithm impractical when focusing on long-term interactions.

2 FINITE-HORIZON GAME

We use the notion of finite-horizon POSGs, or *extensive-form games* (EFGs), to reason about an infinite-horizon pursuit-evasion game. An EFG is a tuple $G = (\mathcal{N}, \mathcal{H}, \mathcal{Z}, \mathcal{T}, u, \mathcal{I})$. $\mathcal{N}$ is the set of players, in our case $\mathcal{N} = \{p, e\}$ where p stands for the pursuer and e for the evader. Set $\mathcal{H}$ denotes a finite set of *histories* of actions taken by all players from the beginning of the game. Every history corresponds to a *node* in the game tree; hence we use the terms history and node interchangeably. Each of the histories may be either (1) *terminal* ($h \in \mathcal{Z} \subseteq \mathcal{H}$) where the game ends and player i gets utility $u_i(h)$, (2) controlled by the nature that selects the successor node according to a fixed probability distribution known to all players, or (3) one of the players from $\mathcal{N}$ may be to act. We consider a zero-sum scenario where $u_p(h) = -u_e(h)$. To simplify the notation we use $u(h)$ to denote pursuer's reward. An ordered list of transitions of player i from root to node h is referred to as a player i's *sequence*. Allowed transitions in the game are modeled using a *transition function* $\mathcal{T}$ that provides a set of successor nodes for each non-terminal history. The imperfect observation of players is modelled via *information sets* $\mathcal{I}_i$ that form a partition over histories h where player $i \in \mathcal{N}$ takes action. We assume perfect recall setting where the players never forget their past actions, i.e. for every $I_i \in \mathcal{I}_i$, all histories $h \in I_i$ have the same player i's sequence. Each information set $I_i \in \mathcal{I}_i$ corresponds to one decision point of player i. A randomized *behavioral strategy* σ_i of player i assigns a distribution over actions to each of the information sets in $\mathcal{I}_i$. σ_i can be represented in the form of a *realization plan* r which assigns probability $r(\sigma_i)$ of playing sequence σ_i to each player i's sequence σ_i. The behavioral strategy at information set $I_i \in \mathcal{I}_i$ reached using a sequence σ_i is then $b(I_i, a) = r(\sigma_i a)/r(\sigma_i)$. A Nash equilibrium (NE) in an EFG is a pair of behavioral strategies, in which each player best-responds the strategy of his opponent. The expected utility of playing NE strategies is termed *value of the game*.

We will now use this terminology to construct an EFG for a finite-horizon version of a pursuit-evasion game with N pursuing units played on a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ for t rounds (we term t as the *horizon*). Part of the game tree is shown in Figure 1. At every round $\tau \leq t$, pursuer's units occupy vertices s_p^τ, where $s_p^\tau = \{s_{p,1}^\tau, \ldots, s_{p,N}^\tau\}$ is an N-element multiset of vertices of $\mathcal{G}$, and the evader is located in vertex $s_e^\tau \in \mathcal{V}$. The goal of the pursuer is to achieve a situation where the evader is caught, i.e. $s_e^\tau \in s_p^\tau$. In every round, players move their units to vertices adjacent to their current positions ($\text{adj}(v)$ denotes the set of vertices adjacent to v). Position of the evader in round $\tau+1$ is thus $s_e^{\tau+1} \in \text{adj}(s_e^\tau)$. We overload the operator adj to apply it also on multisets representing positions of pursuer's units, i.e. $s_p^{\tau+1} \in \text{adj}(s_p^\tau)$, where $\text{adj}(s_p^\tau) = \times_{i=1\ldots N} \text{adj}(s_{p,i}^\tau)$.

A horizon-t game $G^t\langle s_p^0, b^0 \rangle$ is parametrized by the initial position of the pursuer $s_p^0 \in \mathcal{V}^N$ and a distribution over evader's initial positions $b^0 \in \Delta(\mathcal{V})$ known to both players (we term b^0 the initial *belief*). The game starts with a chance move selecting the initial position of the evader s_e^0 (based on b^0).

A history $h \in \mathcal{H}$ in a game with horizon t corresponds to a list of positions $s_e^0 s_p^1 s_e^1 \cdots s_p^\tau s_e^\tau$, where $\tau \leq t$. The utility values are assigned to terminal histories as follows: if the pursuer failed to capture the evader in time, i.e. if $\tau = t$ and $s_e^\tau \notin s_p^\tau$, he gets utility $u(h) = 0$; if he successfully captured the evader in the time limit t, i.e. if $\tau \leq t$ and $s_e^\tau \in s_p^\tau$, he gets the reward

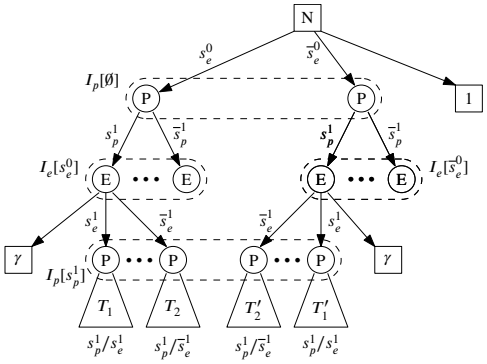

Figure 1: EFG representation of a finite-horizon PEG.

$u(h) = \gamma^{\tau}$ for capturing the evader in τ rounds (where $\gamma \in [0,1)$ is the *discount factor*). The transition function $\mathcal{T}$ complies with the graph (i.e., the adjacency function adj), hence $s_p^{\tau} \in \text{adj}(s_p^{\tau-1})$ and $s_e^{\tau} \in \text{adj}(s_e^{\tau-1})$ for every $\tau \geq 1$. For notational simplicity we denote the sequence of pursuer's actions $s_p^1 \cdots s_p^{\tau}$ in h as $h|_p$ and the sequence of evader's actions $s_e^1 \cdots s_e^{\tau}$ as $h|_e$.

The position of the evader is unknown to the pursuer. Hence, in a perfect recall game, there is one pursuer's information set $I_p[\sigma_p]$ for each of his sequences σ_p where $I_p[\sigma_p] = \{h' \mid h' \in \mathcal{H} \setminus \mathcal{Z} \colon h'|_p = \sigma_p\}$.

Evader, on the other hand, knows the situation almost perfectly. She knows where the pursuer's units were located *before* the pursuer acted in the current round (recall that the pursuer acts first). The only information missing to the evader is the action being taken by the pursuer in the current round. Hence, for every history $h = s_e^0 s_p^1 s_e^1 \cdots s_p^{\tau} s_e^{\tau}$ where the pursuer is to play, there is evader's information set $I_e[h] = \{h s_p^{\tau+1} | s_p^{\tau+1} \in \text{adj}(s_p^{\tau})\}$ containing all possible continuations of the pursuer.

2.1 Shape of the Value Function

The sizes of the extensive-form representation and associated behavioral strategies grow exponentially as the horizon increases. This makes it quickly impossible to use standard algorithms for game trees, especially since we aim to solve *infinite* horizon games.

We alleviate the problem of increasing complexity of the strategy representation by representing strategies *only* using their values. We show that the value of a strategy is linear in the belief, and we can thus represent it using just $|\mathcal{V}|$ real numbers. Moreover, when the horizon is finite, we need to consider only finitely many strategies regardless of the initial belief, which makes value functions, formed by values of best strategies for each belief, be piecewise linear and convex and allows us to represent them compactly.

Definition 1. A *value function* $v^t\langle s_p^0 \rangle : \Delta(\mathcal{V}) \to [0,1]$ is a function assigning the value $v^t\langle s_p^0 \rangle (b^0)$ of the game $G^t\langle s_p^0, b^0 \rangle$ to every initial belief b^0. By v^t we mean a set of value functions $v^t\langle s_p^0 \rangle$, one for each initial position $s_p^0 \in \mathcal{V}^N$ of the pursuer.

In the following text, we show that a value function $v^t\langle s_p^0 \rangle$ is piecewise linear and convex (PWLC) in the belief for every finite horizon t. For notational simplicity, the term linear is used to refer to affine functions as well. The proof is structured as follows: (1) firstly we show that the expected utility of every pursuer's strategy is linear in the belief, next (2) it is sufficient to consider a finite set of pursuer's strategies $\Sigma^t\langle s_p^0 \rangle$ when looking for the Nash equilibrium one; and finally (3) we show that the PWLC nature of the value function follows from (1) and (2).

Lemma 1. *Let σ_p be a randomized behavioral strategy of the pursuer in games $G^t\langle s_p^0, b^0 \rangle$, where the pursuer starts in vertices s_p^0, parametrized by the initial belief b^0. The expected utility of playing σ_p against a best responding opponent is linear in b^0.*

Theorem 1. *Let $G^t\langle s_p^0, b^0 \rangle$ be a horizon-t game parametrized by the initial belief b^0 where the pursuer starts in a set of vertices s_p^0. There exists a finite set of pursuer's behavioral strategies $\Sigma^t\langle s_p^0 \rangle$ such that for every initial belief b^0, there is at least one strategy $\sigma_p \in \Sigma^t\langle s_p^0 \rangle$ that is in Nash equilibrium of $G^t\langle s_p^0, b^0 \rangle$.*

Proof. We use the sequence-form linear program for solving EFGs (Koller et al., 1996) to reason about the set of strategies $\Sigma^t\langle s_p^0 \rangle$. In this LP, values in every information set of the evader, as well as the value $v(root)$ in the root node, are computed in a bottom-up fashion. Every such value $v(I_e)$ of an information set I_e can be seen as a concave piecewise linear function in the space of pursuer's realization plans (a compact representation of his behavioral strategies). The pursuer then seeks for a realization plan that maximizes $v(root)$; the maximizer of which can be found among extreme points of line segments of $v(root)$, i.e. vertices of a polytope bounded by this function (Vanderbei, 2014). We show that the set of such extreme points does not depend on the initial belief b^0.

There is one information set $I_e[s_e^0]$ of the evader for each of her initial positions s_e^0. The utility of every terminal node in the subgame beneath $I_e[s_e^0]$ is multiplied by chance probability $b(s_e^0)$, which allows us to factor out this probability and obtain the following constraint for the root node:

$$v(root) \leq \sum_{s_e^0 \in s_p^0} b^0(s_e^0) + \sum_{s_e^0 \in \mathcal{V} \setminus s_p^0} b^0(s_e^0) \cdot \hat{v}(I_e[s_e^0]) \quad (1)$$

Value $v(root)$ is a convex combination of concave piecewise linear functions $\hat{v}(I_e[s_e^0])$. As the belief was

factored out, these functions, as well as the finite set of their extreme points $P[s_e^0]$, no longer depend on the belief. This convex combination with arbitrary coefficients b^0 cannot have an extreme where none of the functions $\hat{v}(I_e[s_e^0])$ has one. The set of extreme points is therefore a subset of $\bigcup_{s_e^0} P[s_e^0]$ — a finite set that does not depend on the belief. Each of the extreme points in $\bigcup_{s_e^0} P[s_e^0]$ corresponds to one pursuer's realization plan, and thus one his behavioral strategy, which allows us to construct the finite set $\Sigma^t\langle s_p^0\rangle$. □

Theorem 2. *Value function $v^t\langle s_p^0\rangle$ is piecewise linear and convex in the belief space.*

Proof. This result directly follows from Lemma 1 and Theorem 1. There is a finite set of randomized strategies $\Sigma^t\langle s_p^0\rangle$ that has to be considered by the pursuer and value of each such strategy is linear in the belief space. Thus the value function $v^t\langle s_p^0\rangle$ is a pointwise maximum taken over a finite set of linear functions, which is a PWLC function in the belief space. □

A PWLC function can be represented as a finite set of α-*vectors*. Every α-vector $\alpha = (\alpha_1, \ldots, \alpha_{|\mathcal{V}|})$ represents one of the affine functions by assigning an expected reward α_i to each pure belief. We will often work with the α-vector representation of a value function, hence we overload the notation and consider value functions to be sets of such α-vectors as well.

Lemma 1 and Theorem 1 imply that each linear segment of the value function matches one pursuer's strategy, we thus use terms α-vector and pursuer's strategy interchangeably. This is similar to POMDPs where each α-vector matches one conditional plan.

3 VALUE ITERATION

In the previous section, we related the concept of the value functions to the EFG representation of the game and discussed that these functions have desirable properties. We leverage their representation to design a dynamic programming approach inspired by value iteration algorithms for either POMDPs (Smallwood and Sondik, 1973; Monahan, 1982) or perfect information stochastic games (Shapley, 1953). A sequence of value functions $\{v^t\}_{t=0}^{\infty}$ is being constructed by the algorithm, starting with values of a horizon-0 game, where the pursuer wins only when he starts in the same vertex as the evader.

We avoid using the exponentially-sized representation of the underlying EFG by computing value function of a horizon-t game using the solution of the game with horizon $t-1$. First, we state a well-defined value update formula that expresses v^t in terms of v^{t-1} (Theorem 3). We let the players choose their strategies for the first round of the horizon-t game using the maximin principle (we term these *one-step strategies*) and we show that the pursuer can use these strategies to update his belief. Pursuer's one-step strategy π_p is a distribution over possible actions of his units, $\pi_p \in \Delta(\mathrm{adj}(s_p^0))$, from which he samples his action. The evader acts similarly, however she conditions her decision on her *true* position s_e^0 (not just on the overall belief available to the pursuer); her one-step strategy is thus a mapping $\pi_e : \mathcal{V} \to \Delta(\mathcal{V})$, such that $\pi_e(s_e^0)$ assigns zero probability to vertices not adjacent to s_e^0.

The piecewise linearity and convexity of value functions have implications on the computation of the value functions. Firstly it allows finding optimal one-step strategies by means of linear programming (Section 3.1), furthermore, we need not evaluate the value update formula in every point in the belief space to construct new value functions. Instead, we can use an incremental algorithm which inspects extreme points of line segments of a temporary function to check if it can terminate and value function has been constructed, or further linear segments have to be added.

Theorem 3. *The value of $G^t\langle s_p^0, b\rangle$ is computed from value functions v^{t-1} of horizon-$(t-1)$ games. It holds*

$$v^t\langle s_p^0\rangle(b) = \sum_{s_e \in s_p^0} b(s_e) + \tag{2}$$

$$+\gamma\Big[\sum_{s_e \in \mathcal{V}\setminus s_p^0} b(s_e)\Big] \cdot \max_{\pi_p} \min_{\pi_e} \sum_{s_p^1 \in \mathcal{V}^N} \pi_p(s_p^1) \cdot v^{t-1}\langle s_p^1\rangle(b_{\pi_e})$$

where the transformed belief b_{π_e} depends solely on the evader's one-step strategy π_e and the parametrization of the game $G^t\langle s_p^0, b\rangle$:

$$b_{\pi_e}(s_e') = \frac{1}{\sum_{s_e \in \mathcal{V}\setminus s_p^0} b(s_e)} \sum_{s_e \in \mathcal{V}\setminus s_p^0} b(s_e) \cdot \pi_e(s_e, s_e') \tag{3}$$

The computation of v^t using Eq. (2) *forms a dynamic programming operator H, such that $v^t = Hv^{t-1}$.*

Proof. The correctness of the value update formula will be proved by computing the value of $G^t\langle s_p^0, b\rangle$ in a bottom-up fashion. We start by considering that one-step strategies of the players for the first round of the game are fixed, while they play optimally afterward. This determines pursuer's expected reward at every node in the game tree, which we use to express his expected utility in the root node (Lemma 2). As the behavior in the first round of the game is fixed, parts of the game tree are independent on each other — we refer to these subgames as $G[s_p^1]$. This allows us to evaluate the expectation from Lemma 2 by solving these games separately. It turns out that games $G[s_p^1]$ are strategically equivalent to shorter-horizon games

$G^{t-1}\langle s_p^1, b_{\pi_e}\rangle$, the solution of which is represented by value functions v^{t-1}. The expectation can be thus expressed solely in terms of v^{t-1}. Finally, we relax the assumption of fixed strategies in the first round, which yields the desired maximin formula (Equation (2)).

Let $\pi_p \in \Delta(\mathrm{adj}(s_p^0))$ be a fixed pursuer's one-step strategy, and $\pi_e : \mathcal{V} \to \Delta(\mathcal{V})$ be a fixed one-step strategy of the evader. Assume that both players play according to π_p and π_e in the first round of the game, i.e. the pursuer follows π_p in his information set $I_p[\emptyset]$ (i.e. pursuer's information set where he has not acted yet, see Figure 1) and the evader plays according to $\pi_e(s_e^0)$ in her information set $I_e[s_e^0]$ (where she has received the information that she is located in vertex s_e^0). Once the first round of the game is over, players continue with their best strategies available. We denote such optimal strategies where the players are restricted to play π_p and π_e in the first round as σ_p and σ_e.

Definition 2. Let π_p, π_e be fixed one-step strategies for the first round of $G^t\langle s_p^0, b\rangle$ and σ_p, σ_e be optimal strategies with restriction to play π_p and π_e in the first round. The pursuer's expected reward when (σ_p, σ_e) are followed and node h in the game tree is reached is denoted $u(h)$ and termed *expected reward in h*.

We follow by expressing the pursuer's expected utility when strategies (σ_p, σ_e) are followed by propagating expected rewards from subsequent nodes in the game tree. We use histories of the form $s_e^0 s_p^1 s_e^1$ where the evader started in vertex s_e^0 (based on the move of nature) and then in the first round the pursuer moved his units to vertices s_p^1 and the evader moved to s_e^1.

Lemma 2. *The expected reward in the root node is:*

$$u(\emptyset) = \sum_{s_e^0 \in s_p^0} b(s_e^0) + \Big[\sum_{s_e^0 \notin s_p^0} b(s_e^0)\Big] \cdot \sum_{s_p^1} \pi_p(s_p^1) \Big(\gamma \sum_{s_e^1 \in s_p^1} b_{\pi_e}(s_e^1) + \Big[\sum_{s_e^1 \notin s_p^1} b_{\pi_e}(s_e^1)\Big] \cdot \sum_{s_e^1 \notin s_p^1} \sum_{s_e^0 \notin s_p^0} \left[\frac{b(s_e^0) \cdot \pi_e(s_e^0, s_e^1)}{\sum_{\tilde{s}_e^1 \notin s_p^1} \sum_{\tilde{s}_e^0 \notin s_p^0} b(\tilde{s}_e^0) \cdot \pi_e(\tilde{s}_e^0, \tilde{s}_e^1)} \cdot u(s_e^0 s_p^1 s_e^1) \right] \Big) \tag{4}$$

Lemma 2 expressed the value in the root node based on the expected rewards in histories $s_e^0 s_p^1 s_e^1$ where the pursuer is to move. The pursuer knows only s_p^1, hence these histories are partitioned into his information sets $I_p[s_p^1]$, one for each pursuer's move s_p^1 in the first round (see Figure 1). Importantly, for every subgame below $I_p[s_p^1]$, there is no information set that would involve nodes not present in this subgame — neither pursuer nor evader forgets that s_p^1 was played. The optimal behavior in these subgames hence depends only on the belief in $I_p[s_p^1]$, which is fixed due to the fixed behavior in the first round. We can thus compute value of the subgame below $I_p[s_p^1]$ separately by making chance simulate the belief in $I_p[s_p^1]$.

Let us construct a game $G[s_p^1]$ which consists of the information set $I_p[s_p^1]$ and the subgame beneath it. In this game, information set $I_p[s_p^1]$ is reached with probability $\beta = \sum_{s_e^1 \notin s_p^1} b_{\pi_e}(s_e^1)$, while with probability $1-\beta$ the pursuer gets utility γ without play — this accounts for the reward the pursuer gets if he catches the evader in the first round. The nature player simulates the belief $b[s_p^1]$ in the information set $I_p[s_p^1]$, so that the probability of every history in this information set, given this set was reached, is identical with the original game. The value of the game $G[s_p^1]$ corresponds to the following part of the Equation (4):

$$\overbrace{\gamma \sum_{s_e^1 \in s_p^1} b_{\pi_e}(s_e^1)}^{\text{Evader caught in the first round}} + \overbrace{\Big[\sum_{s_e^1 \notin s_p^1} b_{\pi_e}(s_e^1)\Big]}^{\text{Evader not caught in the first round}} \cdot \tag{5}$$

$$\cdot \sum_{s_e^1 \notin s_p^1} \sum_{s_e^0 \notin s_p^0} \underbrace{\left[\frac{b(s_e^0) \cdot \pi_e(s_e^0, s_e^1)}{\sum_{\tilde{s}_e^1 \notin s_p^1} \sum_{\tilde{s}_e^0 \notin s_p^0} b(\tilde{s}_e^0) \cdot \pi_e(\tilde{s}_e^0, \tilde{s}_e^1)} \cdot u(s_e^0 s_p^1 s_e^1) \right]}_{\text{Belief } b[s_p^1] \text{ of history } s_e^0 s_p^1 s_e^1 \text{ in } I_p[s_p^1]}$$

In the case of $G[s_p^1]$, there are multiple histories for every *current* position of the evader s_e^1 in the information set $I_p[s_p^1]$ (resulting from different initial locations of the evader s_e^0). We show that we need not account for different initial positions of the evader, and thus all histories in $I_p[s_p^1]$ with the same current position of the evader s_e^1 can be merged. The resulting game contains a single history for each s_e^1 in $I_p[s_p^1]$, and thus this game is equivalent to a shorter horizon game $G^{t-1}\langle s_p^1, b_{\pi_e}\rangle$ up to multiplication of the utilities by γ to account for a round that has already passed. This allows using the solution of $G^{t-1}\langle s_p^1, b_{\pi_e}\rangle$ represented by value functions v^{t-1} to express the value of $G[s_p^1]$.

Definition 3. Two deterministic game trees over nodes $\mathcal{H}_1, \mathcal{H}_2$ are isomorphic if there exists a bijection $\xi : \mathcal{H}_1 \to \mathcal{H}_2$ such that $v \in \mathcal{H}_1$ is a successor of $u \in \mathcal{H}_1$ if and only if $\xi(v)$ is a successor of $\xi(u)$, $n \in \mathcal{H}_1$ is a pursuer's node if and only if $\xi(n)$ is a pursuer's node, it is a terminal node if and only if $\xi(n)$ is a terminal node and the utilities $u(n) = u(\xi(n))$. Moreover the trees have the same informational structure: two nodes $u, v \in \mathcal{H}_1$ are in the same information set if and only if $\xi(u), \xi(v)$ are in the same information set.

We can observe that subtrees of nodes $s_e^0 s_p^1 s_e^1$ and $\bar{s}_e^0 s_p^1 s_e^1$ (where s_e^0 and $\bar{s}_e^0$ stands for two different initial positions of the evader) are isomorphic as we can

establish a bijection $\xi(s_e^0 s_p^1 s_e^1 h_{rest}) = \bar{s}_e^0 s_p^1 s_e^1 h_{rest}$. The utility of terminal histories does not depend on the initial position of the evader (only on the time the evader was captured). Whenever pursuer's node u is in information set I_p, node $\xi(u)$ is in I_p as well (because pursuer has no way to detect the evader's initial position). Moreover whenever evader cannot distinguish between two histories $s_e^0 s_p^1 s_e^1 \cdots s_p^q$ and $s_e^0 s_p^1 s_e^1 \cdots \bar{s}_p^q$, she cannot distinguish between histories $\bar{s}_e^0 s_p^1 s_e^1 \cdots s_p^q$ and $\bar{s}_e^0 s_p^1 s_e^1 \cdots \bar{s}_p^q$ either (because her uncertainty is related to the pursuer's move at round q, which does not depend on the initial position of the evader). Thus the subtrees have also the same informational structure.

Lemma 3. *Let I be the topmost information set of $G[s_p^1]$ and let belief $b[I]$ over nodes from I be known and fixed. Let $n_1, n_2 \in I$ be two nodes whose subtrees are isomorphic. Then a game G' with the same structure as G with any belief $b'[I]$ in I, satisfying $b[n_1] + b[n_2] = b'[n_1] + b'[n_2]$ and $b[n] = b'[n]$ for all nodes other than n_1 and n_2, has the same value as G.*

Thanks to the Lemma 3 and the isomorphism of the subtrees beneath $s_e^0 s_p^1 s_e^1$ and $\bar{s}_e^0 s_p^1 s_e^1$, histories $s_e^0 s_p^1 s_e^1$ and $\bar{s}_e^0 s_p^1 s_e^1$ can be merged and associated beliefs added up. By repeating this process, we end up with a single history for each current position of the evader s_e^1 (let $s_e^0 s_p^1 s_e^1$ be such history), whose belief is

$$b'[s_p^1](s_e^0 s_p^1 s_e^1) := \frac{\sum_{s_e^0 \notin s_p^0} b(s_e^0) \cdot \pi_e(s_e^0, s_e^1)}{\sum_{\tilde{s}_e^1 \notin s_p^1} \sum_{\tilde{s}_e^0 \notin s_p^0} b(\tilde{s}_e^0) \cdot \pi_e(\tilde{s}_e^0, \tilde{s}_e^1)} \tag{6}$$

$$= \frac{b_{\pi_e}(s_e^1)}{\sum_{\tilde{s}_e^1 \notin s_p^1} b_{\pi_e}(s_e^1)}; \;\; b'[s_p^1](s_e^1) \text{ for short}$$

The updated belief $b'[s_p^1]$ in Equation (6) complies with belief b_{π_e} (Equation (3)) updated with the information that the evader is located in none of the vertices in s_p^1. The belief in $I_p[s_p^1]$ matches the belief in topmost information set of $G^{t-1}\langle s_p^1, b_{\pi_e}\rangle$; and the resulting game is the same as $G^{t-1}\langle s_p^1, b_{\pi_e}\rangle$ up to multiplication by γ. The value of $G[s_p^1]$ (Equation (5)), from which this game was derived, is thus $\gamma v^{t-1}\langle s_p^1\rangle(b_{\pi_e})$. We substitute this value to Equation (4) to obtain

$$u(\emptyset) = \sum_{s_e^0 \in s_p^0} b(s_e^0) + \Big[\sum_{s_e^0 \notin s_p^0} b(s_e^0)\Big] \cdot \cdot \sum_{s_p^1} \pi_p(s_p^1) \cdot \Big(\gamma v^t \langle s_p^1\rangle (b_{\pi_e})\Big) \tag{7}$$

By allowing the players to choose their optimal one-step strategies π_p and π_e in Equation (7), we obtain the desired maximin formula from Equation (2). □

3.1 Computing One-Step Strategies

The evaluation of the Equation (2) involves computation of optimal strategies of the players. In this section we show that if the value functions v^{t-1} are piecewise linear and convex and represented by sets of α-vectors (which holds due to Theorem 2), the strategies can be found out by means of linear programming.

Due to limited space, we provide the linear program for computing optimal one-step strategy in $G^t\langle s_p^0, b\rangle$ for the pursuer only. At the beginning of each round, the pursuer realizes what vertices the evader is *not* located in, and hence updates his belief about the position of the evader. We thus restrict ourselves to the case where $b(s_e) = 0$ for all $s_e \in s_p^0$.

In the following linear program, the pursuer seeks for a strategy maximizing his expected utility against the best-responding opponent. He assumes strategies of the form "move to s_p^1 first and then follow strategy whose value is represented by $\alpha \in v^{t-1}\langle s_p^1\rangle$". The choice of α uniquely defines such strategy. The probability of playing each strategy $\alpha \in v^{t-1}\langle s_p^1\rangle$ is represented by variable $\hat{\pi}_p(s_p^1, \alpha)$. Constraint (9) corresponds to the value of playing such randomized strategy against the best-responding evader who starts in vertex s_e ($\alpha(s_e')$ denotes the value of α evaluated at pure belief corresponding to action s_e' of the evader). The evader starts in s_e with probability $b(s_e)$, hence the objective (8) calculates the expectation over individual $v(s_e)$. For the resulting one-step strategy of the pursuer, it holds that $\pi(s_p^1) = \sum_{\alpha \in v^{t-1}\langle s_p^1\rangle} \hat{\pi}(s_p^1, \alpha)$.

$$\max_{v, \hat{\pi}_p} \gamma \sum_{s_e \in \mathcal{V}} b(s_e) \cdot v(s_e) \tag{8}$$

$$\text{s.t.} \sum_{s_p^1 \in \text{adj}(s_p^0)\,;\, \alpha \in v^{t-1}\langle s_p^1\rangle} \alpha(s_e') \cdot \hat{\pi}_p(s_p^1, \alpha) \geq v(s_e) \quad \forall \{s_e, s_e'\} \in \mathcal{E} \tag{9}$$

$$\sum_{s_p^1 \in \text{adj}(s_p^0)\,;\, \alpha \in v^{t-1}\langle s_p^1\rangle} \hat{\pi}_p(s_p^1, \alpha) = 1 \tag{10}$$

$$\hat{\pi}_p(s_p^1, \alpha) \geq 0 \quad \forall s_p^1 \in \text{adj}(s_p^0)\; \forall \alpha \in v^{t-1}\langle s_p^1\rangle \tag{11}$$

3.2 Computing Value Functions

In each iteration of our value iteration algorithm, value functions v^t are constructed from the solution from the previous iteration — value functions v^{t-1}. By repeating this construction, a sequence of finite-horizon value functions $\{v^t\}_{t=0}^{\infty}$ approaching the values of the infinite-horizon game is being constructed. The value functions v^t to be constructed, as well as v^{t-1}, are PWLC (Theorem 2). We show that this allows us to avoid evaluating the dynamic programming

operator H (Equation (2)) in every point in the belief space and enables us to construct v^t using only a finite subset of beliefs; the extreme points of line segments of v^t. We proceed in two steps: (1) first, we compute a function $Q^t_{\pi_p}\langle s^0_p\rangle$ corresponding to the expected utility the pursuer gets if he plays π_p at the first round of the longer horizon game $G^t\langle s^0_p, b\rangle$; (2) then we show how to compute $v^t\langle s^0_p\rangle$ as a combination of multiple $Q^t_{\pi_p}\langle s^0_p\rangle$ for properly chosen one-step strategies π_p. We start with a formal definition of function $Q^t_{\pi_p}\langle s^0_p\rangle$.

Definition 4. Let π_p be pursuer's one-step strategy for the first round of the game $G^t\langle s^0_p, b\rangle$. The *value of* π_p is a function $Q^t_{\pi_p}\langle s^0_p\rangle$ assigning the expected reward the pursuer gets in the game $G^t\langle s^0_p, b\rangle$ against the best-responding opponent, when he plays π_p in the first round and continues by playing according to his optimal strategy in the rest of the game, i.e.

$$Q^t_{\pi_p}\langle s^0_p\rangle(b) := \sum_{s_e \in s^0_p} b(s_e) + \gamma\Big[\sum_{s_e \in \mathcal{V}\setminus s^0_p} b(s_e)\Big] \cdot \cdot \min_{\pi_e} \sum_{s^1_p \in \mathcal{V}^N} \pi_p(s^1_p) \cdot v^{t-1}\langle s^1_p\rangle(b_{\pi_e}) \tag{12}$$

According to the previous definition, once the first round of the game is over, the pursuer continues with his optimal strategy. The following lemma shows that this optimal strategy for the rest of the game can be characterized by α-vectors of v^{t-1}.

Lemma 4. *Let π_p be pursuer's fixed one-step strategy for the first round of the game. For every belief b there are strategies $\sigma_p[s^1_p]$, one for each $s^1_p \in \mathrm{adj}(s^0_p)$, represented by α-vectors $\alpha[s^1_p] \in v^{t-1}\langle s^1_p\rangle$, such that it is optimal to follow $\sigma_p[s^1_p]$ when s^1_p was played in the first round of the game. The value of strategy σ_p prescribing the pursuer to play according to π_p in the first round and continue by using respective $\sigma_p[s^1_p]$ is linear and the corresponding α-vector satisfies*

$$\alpha^{\sigma_p}(s_e) = \begin{cases} 1 & s_e \in s_p \\ \gamma \min\limits_{s'_e \in \mathrm{adj}(s_e)} \sum\limits_{s^1_p} \pi_p(s^1_p) \cdot \alpha[s^1_p](s'_e) & \text{otherwise} \end{cases} \tag{13}$$

Lemma 4 gives us a direct algorithm for computing $Q^t_{\pi_p}$. PWLC functions v^{t-1} correspond to a finite number of horizon-t strategies, represented by a finite number of α-vectors. There is only a finite number of ways to choose strategies $\sigma_p[s^1_p]$ from Lemma 4, which can be found by means of enumeration. The maximization over linear functions representing value of such strategies corresponds to the function $Q^t_{\pi_p}\langle s_p\rangle$ which is thus piecewise linear and convex.

The definition of $Q^t_{\pi_p}\langle s_p\rangle$ implies that we can compute the value function $v^{t+1}\langle s_p\rangle$ by allowing the pursuer to play arbitrary strategy π_p, when

$$\hat{v}^t\langle s^0_p\rangle \leftarrow \left\{\mathbf{0}^{|\mathcal{V}|}\right\},\ \hat{\Pi}_p \leftarrow \emptyset$$
while $\exists b \in \Delta(\mathcal{V}) : v^t\langle s^0_p\rangle(b) > \hat{v}^t\langle s^0_p\rangle(b)$ **do**
  $\pi_p \leftarrow$ optimal strategy of the pursuer at belief b for the first round (see (8))
  $\hat{\Pi}_p \leftarrow \hat{\Pi}_p \cup \{\pi_p\}$
  $\hat{v}^t\langle s^0_p\rangle \leftarrow \hat{v}^t\langle s^0_p\rangle \oplus Q^t_{\pi_p}\langle s_p\rangle$
return $\hat{v}^t\langle s_p\rangle$

Algorithm 1: Incremental construction of $v^t\langle s_p\rangle$.

$$v^t\langle s^0_p\rangle(b) = \max_{\pi_p} Q^t_{\pi_p}\langle s^0_p\rangle(b) \tag{14}$$

As a consequence of Theorem 1, it is sufficient to consider a finite set Π_p of strategies in the maximizer of Equation (14) and obtain $v^t\langle s^0_p\rangle$ as the pointwise maximum from respective $Q^t_{\pi_p}\langle s^0_p\rangle$ functions, $v^t\langle s_p\rangle = \bigoplus_{\pi_p \in \Pi_p} Q^t_{\pi_p}\langle s_p\rangle$. The set of such strategies is however initially unknown. We propose the Algorithm 1 that constructs both the set of strategies $\hat{\Pi}_p$ and the value function $\hat{v}^t\langle s^0_p\rangle$ incrementally by iteratively verifying whether the current set $\hat{\Pi}_p$ is sufficient for obtaining the actual value function $v^t\langle s^0_p\rangle$.

The algorithm constructs a set of strategies $\hat{\Pi}_p$ and a corresponding estimate of value function $\hat{v}^t\langle s^0_p\rangle$, starting with empty $\hat{\Pi}_p$. In each iteration, it verifies if strategies $\hat{\Pi}_p$ used to form current $\hat{v}^{t+1}\langle s^0_p\rangle$ are optimal in every belief $b \in \Delta(\mathcal{V})$. If a belief b where the strategy can be improved is found, i.e. $Q^t_{\pi_p}\langle s^0_p\rangle(b) > \hat{v}^t\langle s^0_p\rangle(b)$ for some π_p, it updates $\hat{\Pi}_p$ and recomputes $\hat{v}^t\langle s_p\rangle$. If no such belief b exists, all required strategies were considered and $\hat{v}^t\langle s^0_p\rangle = v^t\langle s^0_p\rangle$.

Whenever the value function $\hat{v}^t\langle s^0_p\rangle$ is not yet optimal for all beliefs, i.e. there exists a belief b where $v^t\langle s^0_p\rangle(b) > \hat{v}^t\langle s^0_p\rangle(b)$, there exists a belief b' with the same property that forms an extreme point of a line segment on $\hat{v}^t\langle s^0_p\rangle$. This is characterized by Lemma 5.

Lemma 5. *If there is a belief b where $v^t\langle s^0_p\rangle(b) > \hat{v}^t\langle s^0_p\rangle(b)$, there must be a belief b' that forms an extreme point of a line segment on the surface of $\hat{v}^t\langle s^0_p\rangle$ where $v^t\langle s^0_p\rangle(b') > \hat{v}^t\langle s^0_p\rangle(b')$.*

Thanks to Lemma 5, we can consider only a *finite* set of beliefs that form extreme points of line segments on the value function $\hat{v}^t\langle s^0_p\rangle$. In every iteration, a one-step strategy that is optimal at some belief point (and thus must be present in Π_p) is added to $\hat{\Pi}_p$. Due to Theorem 1, the set Π_p required to obtain the optimal value function $v^t\langle s^0_p\rangle$ is finite. Hence after a finite number of iterations, the Algorithm 1 terminates.

3.3 Convergence of the Algorithm

We demonstrate the convergence of our value iteration algorithm by showing that the dynamic programming operator H (Equation 2) has a *unique* fixpoint which is reached by its iterative application. We obtain this by showing that H is a contraction mapping under the following max-norm and applying the Banach's fixed point theorem (Ciesielski et al., 2007).

$$\|v - \bar{v}\| = \max_{s_p^0 \in \mathcal{V}^N} \max_{b \in \Delta(\mathcal{V})} |v\langle s_p^0 \rangle(b) - \bar{v}\langle s_p^0 \rangle(b)| \quad (15)$$

Lemma 6. *The operator H is a contraction with contractivity factor $\gamma < 1$ under max-norm.*

Theorem 4. *There is a unique set of value functions v^* satisfying $v^* = Hv^*$ and the recursive application of H converges to v^*. Series $\{v^t\}_{i=0}^{\infty}$ thus converges to value functions of an infinite horizon game.*

Proof. The operator H is a contraction mapping defined on a metric space of sets of bounded functions defined on the belief space. By applying Banach's fixed point theorem (Ciesielski et al., 2007) we get that H has a unique fixed point v^* and the recursive application of H converges to v^*. □

Proposition 1. *After t iterations of the value iteration algorithm it holds that $\|v^t - v^*\| \leq \gamma^t$.*

4 CONCLUSIONS

We present the first algorithm for solving the class of two-player discounted pursuit-evasion games with infinite horizon and partial observability, where the evader is assumed to be perfectly informed about the current state of the game (i.e. position of pursuer's units). This class of games has a significant relevance in security domains where a robust strategy that provides guarantees in the worst case is often desirable.

Our algorithm is a modification of the well-known value iteration algorithm for solving Partially Observable Markov Decision Processes (POMDPs), or stochastic games with concurrent moves. We show that the strategies can be compactly represented using value functions that depend on the location of the pursuing units and the belief about the position of the evader, but not explicitly on the history of moves. These value functions are piecewise linear and convex and allow us to design a dynamic programming operator for the value iteration algorithm.

Our work is the first step towards many practical algorithms for solving discounted stochastic games with one-sided partial observability. These can be applied in many scenarios requiring robust strategies and thus our work opens the whole new area of research in algorithmic and computational game theory. One natural continuation is an adaptation of point-based approximation algorithms for POMDPs to improve the scalability of the value iteration algorithm.

ACKNOWLEDGEMENTS

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

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

Hansen, E. A., Bernstein, D. S., and Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pages 709–715.

Koller, D., Megiddo, N., and Von Stengel, B. (1996). Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259.

McEneaney, W. M. (2004). Some classes of imperfect information finite state-space stochastic games with finite-dimensional solutions. *Applied Mathematics and Optimization*, 50(2):87–118.

Monahan, G. E. (1982). State of the arta survey of partially observable Markov decision processes: theory, models, and algorithms. *Management Science*, 28(1):1–16.

Pineau, J., Gordon, G., Thrun, S., et al. (2003). Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*, volume 3, pages 1025–1032.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100.

Smallwood, R. D. and Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088.

Smith, T. and Simmons, R. (2012). Point-based POMDP algorithms: Improved analysis and implementation. *arXiv preprint arXiv:1207.1412*.

Vanderbei, R. J. (2014). *Linear programming*. Springer.

Vidal, R., Shakernia, O., Kim, H. J., Shim, D. H., and Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669.