Solving Partially Observable Stochastic Shortest-Path Games

Petr Tomášek[1*] , **Karel Horák**[1] , **Aditya Aradhye**[1] ,
Branislav Bošanský[1] and **Krishnendu Chatterjee**[2]
[1]Artificial Intelligence Center, Dept. of Computer Science
Faculty of Electrical Engineering, Czech Technical University in Prague
[2]Institute of Science and Technology Austria
{petr.tomasek, karel.horak, aditya.aradhye, branislav.bosansky}@aic.fel.cvut.cz,
krishnendu.chatterjee@ist.ac.at

Abstract

We study the two-player zero-sum extension of the partially observable stochastic shortest-path problem where one agent has only partial information about the environment. We formulate this problem as a partially observable stochastic game (POSG): given a set of target states and negative rewards for each transition, the player with imperfect information maximizes the expected undiscounted total reward until a target state is reached. The second player with the perfect information aims for the opposite. We base our formalism on POSGs with one-sided observability (OS-POSGs) and give the following contributions: (1) we introduce a novel heuristic search value iteration algorithm that iteratively solves depth-limited variants of the game, (2) we derive the bound on the depth guaranteeing an arbitrary precision, (3) we propose a novel upper-bound estimation that allows early terminations, and (4) we experimentally evaluate the algorithm on a pursuit-evasion game.

1 Introduction

Stochastic shortest path (SSP) problem [Bertsekas and Tsitsiklis, 1991; Bertsekas, 1995] belongs to classical problems in which an agent aims to find an optimal plan to reach a target state in a stochastic environment (a problem with *indefinite horizon*). The problem can be modeled as a Markov Decision Process (MDP) where the objective is an undiscounted sum of rewards[1] – e.g., for each transition, the agent receives a penalty, and thus the agents want to reach the target state where no additional penalty is received. This model naturally arises in many fields including robotics [Lim *et al.*, 2013; Saisubramanian *et al.*, 2019], wireless networks [Chen *et al.*, 2007] or model checking [Norman *et al.*, 2005].

Throughout the years, many variants of the original problem have emerged. We focus on two of them with significant practical implications: (i) partially observable SSP [Patek, 1999; Egorov *et al.*, 2016; Horák *et al.*, 2018; Delamer *et al.*, 2019] and (ii) planning in the presence of an adversary and SSP games [Patek and Bertsekas, 1999; Neu *et al.*, 2012; Rosenberg and Mansour, 2020; Chen *et al.*, 2020]. The partially observable SSP (also referred to as Goal-POMDP) generalizes the model to better reflect real-world scenarios where perfect information is not always available (e.g., robotic sensors are imprecise, and the true position of the robot in an environment may be unknown). For the latter variant, formulating the problem as a game against an opponent allows the agent to find robust plans in an adversarial environment.

Until now, however, the combination of these two variants – an SSP game with partial observability – has not been sufficiently covered by the existing works. We address this gap and, to the best of our knowledge, solve *Partially Observable Stochastic Shortest-Path Games* (POSSPGs) for the first time. We focus on the two-player zero-sum setting. From the game-theoretic perspective, a POSSPG is a variant of Partially Observable Stochastic Games (POSGs) that are intractable in general. One of the main reasons for the intractability of POSGs is reasoning about uncertainty – one player must consider a belief over the possible states of the environment, but also opponent's beliefs, and also beliefs over beliefs, and a belief hierarchy in general.

To avoid such nesting, we restrict to a simplified setting where one player has perfect information about the course of the game. In POSSPG, this is quite natural since we can assume that the adversary has the perfect information and the agent seeks for a robust plan against a well-informed opponent. Moreover, the existing works on POSGs with such an information asymmetry (termed one-sided POSGs [Horák *et al.*, 2017]; OS-POSGs) offer algorithms for solving such games [Horák *et al.*, 2017; Horák *et al.*, 2020]. Unfortunately, the heuristic search value iteration (HSVI) algorithm introduced for one-sided POSGs uses the discounted-sum objective, and the proof of the convergence of the algorithm relies on the discount factor being strictly smaller than 1. Therefore, the existing algorithm cannot be used for POSSPGs where agents maximize the undiscounted sum of rewards.

We address these challenges and introduce a new algorithm for solving POSSPGs based on HSVI. Although the high-level idea of our algorithm is simple, its realization that yields theoretical guarantees in the game-theoretic context is non-trivial. In our algorithm, we solve a sequence of depth-limited variants of the original game (termed *k-cutoff games*) while

[*] Contact Author

[1] Throughout the paper we assume that agents maximize their rewards (i.e., a cost/penalty means a negative reward for an agent).

gradually increasing the depth limit. The main challenge stems from determining whether the currently found solution is sufficiently close to the optimum or whether the algorithm should continue with an increased depth limit. To this end, we (i) derive a theoretical bound on the depth-limit guaranteeing desired precision ε and (ii) extend HSVI with an auxiliary upper-bound estimation of the value of the original unbounded game. This auxiliary upper bound then allows earlier terminations if it is sufficiently close to the lower bound from the k-cutoff games. We evaluate the performance of our algorithm on a pursuit-evasion game on a graph and demonstrate that it is not feasible to use the original HSVI for OS-POSGs with very large discount factors as an alternative to our new HSVI algorithm for POSSPGs.

2 One-sided Partially Observable Stochastic Games (OS-POSG)

We start by reviewing the closely related model of one-sided partially observable stochastic games (OS-POSGs) [Horák *et al.*, 2017; Horák *et al.*, 2020]: these are two-player zero-sum infinite-horizon games played on a graph, where one of the players (player 1, P1) is imperfectly informed about the course of the game, while his adversary (player 2, P2) is perfectly informed. Unlike our proposed model, however, the objective of the players is to optimize the *discounted sum* of rewards obtained in the course of the game.

Formally, an OS-POSG G is a tuple $G = \langle S, A_1, A_2, O, T, R, b_{\text{init}}\rangle$. The game starts in one of the states $s^{(0)} \in S$, where S is the finite set of states. The state $s^{(0)}$ is sampled from the probability distribution over states $b_{\text{init}} \in \Delta(S)$ called the *initial belief.* Then, in each stage t of the game ($t \geq 1$), the players choose their actions $a_1^{(t)} \in A_1$ and $a_2^{(t)} \in A_2$ *simultaneously* and *independently* on each other. Based on their choice, the game transitions to a new state $s^{(t)}$ and an observation $o^{(t)} \in O$ is generated for P1. The set O contains all possible observations for P1 that provide P1 with the partial information about the new state. Formally, the transition function T ensures that $s^{(t)}$ and $o^{(t)}$ are generated with probability $T(o^{(t)}, s^{(t)} \mid s^{(t-1)}, a_1^{(t)}, a_2^{(t)})$. For this transition, P1 receives the stage reward $R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})$. P2 can use the entire past history of the game $s^{(0)}(a_1^{(t')}, a_2^{(t')}, o^{(t')}s^{(t')})_{t'=1}^{t-1}$ to make his decision about his upcoming action $a_2^{(t)}$. P1, on the other hand, can only consider his own past actions and observations $(a_1^{(t')}, o^{(t')})_{t'=1}^{t-1}$ when making his decision. Total expected utility for P1 is defined as the expected infinite discounted sum of stage rewards, that is $\mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})]$. The discount factor γ is assumed to be positive and strictly less than 1. The goal of P1 is to maximize his total expected utility, while P2 aims at minimizing total expected utility of P1.

2.1 Solving OS-POSGs

One of the options to solve OS-POSGs is to (approximately) find the optimal value function $V^* : \Delta(S) \to \mathbb{R}$ of the game G. This function maps beliefs $b \in \Delta(S)$ of P1 to the discounted payoff $V^*(b)$ he can achieve in G given that b is the current distribution over possible states of the game and it can be characterized by a Bellman-style fixed point equation [Horák *et al.*, 2017]:

$$V^*(b) = [HV^*](b) = \max_{\pi_1} \min_{\pi_2} \Big[\mathbb{E}_{b,\pi_1,\pi_2}[R(s, a_1, a_2)] + \\ + \gamma \sum_{a_1,o} \mathbb{P}_{b,\pi_1,\pi_2}[a_1, o] \cdot V^*(\tau(b, a_1, \pi_2, o))\Big] \tag{1}$$

where $\pi_1 : \Delta(A_1)$ and $\pi_2 : S \to \Delta(A_2)$ are stage strategies player 1 and 2 use to choose their actions a_1 and a_2 in the current stage of the game and $\tau(b, a_1, \pi_2, o) = \mathbb{P}_{b,\pi_2}[s' \mid a_1, o]$ is the Bayesian belief update.

Similarly to the single-player case (POMDP), V^* can be approximated using more scalable algorithms. For OS-POSGs, there exists a modification of the heuristic search value iteration algorithm (HSVI) [Horák *et al.*, 2017] that extends HSVI [Smith and Simmons, 2004; Smith and Simmons, 2005] to the two-player zero-sum games. This algorithm (Algorithm 1) uses a self-play between P1 and P2 to identify relevant beliefs in the game and aims at achieving a close approximation of V^* primarily in these relevant beliefs. Piecewise linear and convex lower $\underline{V}$ and upper $\overline{V}$ bounds on V^* are maintained and gradually refined over time. Since V^* is δ-Lipschitz continuous for $\delta = (\max R(\cdot) - \min R(\cdot))/2(1-\gamma)$, δ-Lipschitz bounds $\underline{V}$ and $\overline{V}$ are considered.

The goal of the algorithm is to ensure that a desired precision $\overline{V}(b_{\text{init}}) - \underline{V}(b_{\text{init}}) \leq \varepsilon$ is eventually reached (line 1). To this end, the algorithm performs a sequence of trials (represented by the Explore procedure) that aim at achieving that beliefs $b^{(k)}$ reached at k-th level of recursion are approximated within $\rho(k)$ precision. Sequence ρ is defined as

$$\rho(1) = \varepsilon \qquad\qquad \rho(k+1) = [\rho(k) - 2\delta D]/\gamma\ . \tag{2}$$

where δ is the Lipschitz constant of V^* and $D > 0$ is a parameter such that ρ is monotonically increasing and unbounded.

In each stage of a trial, players choose their stage strategies $\pi_1^{(k)}$ and $\pi_2^{(k)}$ (lines 5–6). The maximizing player 1 chooses his strategy based on solving the game corresponding to $[H\overline{V}](b)$, while the minimizing player 2 chooses his strategy based on solving $[H\underline{V}](b)$ (i.e., each player obtain his own strategy based upon reasoning about *optimistic* variant of the stage game from his perspective).[2]

Each trial constructs a sequence of beliefs (line 7) such that (1) $b^{(k+1)}$ is reachable from $b^{(k)}$ when employing stage strategies $\pi_1^{(k)}$ and $\pi_2^{(k)}$, (2) $b^{(k+1)}$ has the highest excess gap

$$\text{excess}_{k+1}(\tau(b^{(k)}, a_1, \pi_2^{(k)}, o)) = \\ = \overline{V}(\tau(b^{(k)}, a_1, \pi_2^{(k)}, o)) - \underline{V}(\tau(b^{(k)}, a_1, \pi_2^{(k)}, o)) - \rho(k) \tag{3}$$

among the reachable beliefs weighted by the probability $\mathbb{P}_{b^{(k)},\pi_1^{(k)},\pi_2^{(k)}}[a_1, o]$ of reaching that belief. Upon reaching a belief $b^{(k)}$ where sufficient accuracy is achieved (i.e.,

[2] [Horák *et al.*, 2020] shows that for piecewise linear and convex bounds $\underline{V}$ and $\overline{V}$ these strategies can be obtained by means of linear programming.

Algorithm 1: `HSVI` for discounted `OS-POSGs`

```
1 while $\overline{V}(b_{\text{init}}) - \underline{V}(b_{\text{init}}) > \varepsilon$ do
2     Explore($b_{\text{init}}$, 1)
3 procedure Explore($b^{(t)}$, $t$)
4     if $\text{excess}_t(b^{(t)}) \le 0$ then return
5     $\pi_1^{(t)} \leftarrow$ optimal strategy of P1 in $[H\overline{V}](b^{(t)})$
6     $\pi_2^{(t)} \leftarrow$ optimal strategy of P2 in $[H\underline{V}](b^{(t)})$
7     $b^{(t+1)} \leftarrow \arg\max_{a_1,o} \mathbb{P}_{b^{(t)},\pi_1^{(t)},\pi_2^{(t)}}[a_1,o] \cdot$
        $\text{excess}_{t+1}(\tau(b^{(t)}, a_1, \pi_2^{(t)}, o))$
8     Explore($b^{(t+1)}$, $t+1$)
9     Perform point-based update of $\underline{V}$ and $\overline{V}$ in $b^{(t)}$
```

$\text{excess}_k(b^{(k)}) \le 0$), the trial terminates and a sequence of point-based updates in beliefs $b^{(1)}, \dots, b^{(k)}$ is performed on line 9 to refine the bounds $\underline{V}$ and $\overline{V}$ (the updates ensure that $\underline{V}(b) := [H\underline{V}](b)$ and $\overline{V}(b) := [H\overline{V}](b)$ for selected beliefs b).

Convergence properties When discussing our proposed modifications of the `HSVI` algorithm, we will refer to key properties of `HSVI` algorithm for `OS-POSG` that suffice to ensure the convergence:

(1) *Each trial terminates in* $b^{(k)}$ *with* $\text{excess}_k(b^{(k)}) \le 0$ *for some* k. The payoff in the game is bounded and so is the gap between $\underline{V}$ and $\overline{V}$. As ρ is unbounded, $\rho(k)$ will eventually exceed $\overline{V}(b^{(k)}) - \underline{V}(b^{(k)})$. Observe that this also means that all beliefs reachable from $b^{(k-1)}$ have negative excess gap on line 4.

(2) *Point-based update in* $b^{(k-1)}$ *ensures that all beliefs within hypersphere with radius D centered in* $b^{(k-1)}$ *have negative excess gap.* [Horák *et al.*, 2017, Lemma 4]

(3) *Eventually, no beliefs with* $\text{excess}_k(b^{(k)}) > 0$ *remain (unless the algorithm converges earlier).* By (1) and (2), each trial identifies (at least) one belief $b^{(k-1)}$ with positive excess gap and makes all beliefs within D-neighborhood of $b^{(k-1)}$ have negative excess gap. By a standard packing argument, the hyperspheres with radius D describing beliefs with negative excess gap eventually cover entire belief space, hence no beliefs with positive excess gap remain. [Horák *et al.*, 2017, Theorem 3]

Observe that the absence of beliefs with negative excess gap means that $\text{excess}_1(b_{\text{init}}) < 0$ and hence $\overline{V}(b_{\text{init}}) - \underline{V}(b_{\text{init}}) \le \rho(1) = \varepsilon$.

3 Partially Observable Stochastic Shortest-Path Games (POSSPGs)

Partially observable stochastic shortest-path games (`POSSPGs`) extend the partially observable version of Stochastic Shortest-Path problem to a game with the presence of a well-informed adversary. The goal of the player 1 is to reach one of the states in the set S_G of goal states with minimal cost, while the adversary tries to prevent reaching S_G or at least render the cost for reaching S_G as high as possible. We formalize `POSSPGs` based on the formalism of one-sided `POSGs` (Section 2) as a tuple $G = \langle S, A_1, A_2, O, T, R, b_{\text{init}}, S_G \rangle$. Unlike `OS-POSGs` where the objective was the maximization of discounted sum of rewards, the objective of `POSSPGs` is the maximization of total sum of rewards (for the reasons of notational consistency with prior work on `OS-POSGs`, we use negative rewards instead of positive costs):

$$U_b^{\infty}(\sigma_1, \sigma_2) = \mathbb{E}_{b,\sigma_1,\sigma_2}\left[\sum_{t=1}^{\infty} R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})\right] \tag{4}$$

where b is the current belief of the game, and (σ_1, σ_2) are the strategies used by the players. We aim for the worst-case and assume that player 2 is perfectly informed (hence his strategy $\sigma_2 : S(A_1A_2OS)^* \to \Delta(A_2)$ conditions on the entire history of the game), while player 1 who wants to reach S_G has limited observability of the game ($\sigma_1 : (A_1O)^* \to \Delta(A_1)$). Similarly to `OS-POSGs`, we use $V^*(b)$ to denote the value of `POSSPG` with initial belief b (i.e., the payoff the players are guaranteed to achieve under optimal strategies). We also use the notation $\text{val}^{\sigma_1}(b)$ to denote the payoff strategy σ_1 of player 1 achieves in a game with initial belief b (i.e. $\text{val}^{\sigma_1}(b) = \min_{\sigma_2} U_b^{\infty}(\sigma_1, \sigma_2)$).

Since the undiscounted total-sum payoff can be infinite in general, we impose the following assumption in `POSSPGs`:

- *The goal of player 1 is to reach* $s \in S_G$. The game ends[3] upon reaching a state in S_G, and player 1 incurs no further costs (i.e., $R(s, \cdot) = 0$ for all $s \in S_G$). All other costs are strictly positive (i.e., $R(s, \cdot) \le \overline{R} < 0$ for all $s \notin S_G$). This ensures that any strategy σ_1 with finite value val^{σ_1} reaches S_G almost surely.
- *Uniform strategy* σ^{unif} *of player 1 has finite value* $\text{val}^{\sigma^{\text{unif}}} > -\infty$. In other words, player 2 cannot prevent rational player 1 from reaching S_G and the value of the game is finite since $-\infty < \text{val}^{\sigma^{\text{unif}}}(b) \le \max_{\sigma_1} \text{val}^{\sigma_1}(b) = V^*(b) \le 0$.

Although `POSSPGs` are similar to `OS-POSGs`, using discount factor $\gamma = 1$ and applying Algorithm 1 designed for discounted problems to solve `POSSPGs` is impossible. This is primarily based on the fact that the convergence results for Algorithm 1 rely on the γ-contractivity of the backup operator H (Equation 1). [Horák *et al.*, 2018] provides further evidence supporting our claim by showing counterexamples illustrating that the standard `HSVI` algorithm for discounted problems cannot be applied to `Goal-POMDPs`. Since `POSSPG` is an extension of the `Goal-POMDP` model, these results naturally apply.

To this end, we propose a novel algorithm that circumvents these issues by solving a finite-horizon variant of `POSSPGs`, and we show that this allows us to obtain a close approximation of the solution of an infinite-horizon `POSSPG`.

3.1 k-cutoff Game

Similarly to [Horák *et al.*, 2018], we can aim at solving a finite variant of the problem. We term this finite-horizon variant a *k-cutoff game*, and we show that this game can be used as an approximation of an infinite-horizon `POSSPG`.

[3] We model this by setting $T(o_{\text{reach}}, s \mid s, \cdot) = 1$ for every $s \in S_G$ where $o_{\text{reach}} \in O$ informs player 1 about reaching the goal.

A k-cutoff game (denoted G_k) models a decision-making problem where player 1 can play arbitrarily in the first k stages of POSSPG G and is forced to follow uniform strategy σ_{unif} in the rest of the game. This corresponds to a finite-horizon POSG associated with G with objective

$$\begin{aligned} U_b^k(\sigma_1,\sigma_2) = \mathbb{E}_{b,\sigma_1,\sigma_2}\Big[\sum_{t=1}^{k} R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)}) + \\ + \mathrm{val}^{\sigma_{\text{unif}}}(b^{(k+1)})\Big] \end{aligned} \tag{5}$$

where $b^{(k+1)}$ stands for the belief of player 1 after k-th stage of the game. Observe that in the case the game terminated within first k stages, player 1 has been notified of the termination by o_{reach} observation and $\mathsf{Supp}(b^{(k+1)}) \subseteq S_G$, hence value of uniform strategy $\mathrm{val}^{\sigma_{\text{unif}}}(b^{(k)}) = 0$. Similarly to the infinite-horizon case, we define $V_k^*(b)$ as the value of k-cutoff game starting in belief b, and we use $\mathrm{val}_k^{\sigma_1}(b) = \min_{\sigma_2} U_b^k(\sigma_1,\sigma_2)$ to refer to the value of strategy σ_1.

We now formally prove that k-cutoff game G_k can be used to approximate the solution of infinite-horizon POSSPG G. Namely, we show that (1) value of G_k forms a lower bound on the value of G, and (2) we can choose k to make the approximation of value of G arbitrarily tight.

Proposition 1. *Let $k \geq 1$ and $b \in \Delta(S)$ be arbitrary. Then* $\mathrm{val}^{\sigma_{\text{unif}}}(b) \leq V_k^*(b) \leq V^*(b)$.

Proof. Let σ_1^{k*} be an optimal strategy for player 1 in G_k and let σ_1^i be the strategy of player 1 in G which imitates σ_1^{k*} for first k stages and plays uniformly afterwards. Note that the game G_k is played for the k stages only. After the cutoff, the game ends, and the player 1 pays the one-time penalty associated with playing the uniform strategy from the current belief (see the $\mathrm{val}^{\sigma_{\text{unif}}}(b^{(k+1)})$ term in Equation (5)). So, σ_1^{k*} defines the behaviour of player 1 only for k stages. On the other hand, as G is an infinite horizon game, σ_1^i defines the behavior of player 1 in the infinite setting. In this strategy, the decision to play uniformly after the k stages is a deliberate choice of the player (unlike in the case of σ_1^{k*} in G_k where the uniform play is part of the game rules).

As σ_1^{k*} is an optimal strategy for player 1 in G_k, we have $V_k^*(b) = \mathrm{val}_k^{\sigma_1^{k*}}(b)$. As σ_1^i in G imitates σ_1^{k*} for first k stages and plays uniformly afterwards, we argue that these two strategies yield the same value in their respective games. Hence $\mathrm{val}^{\sigma_1^i}(b) = \mathrm{val}_k^{\sigma_1^{k*}}(b)$. We have $V^*(b) = \mathrm{val}^{\sigma_1^*}(b) \geq \mathrm{val}^{\sigma_1^i}(b) = \mathrm{val}_k^{\sigma_1^{k*}}(b) = V_k^*(b)$ where σ_1^* is an optimal strategy for player 1 in G. Thus, $V^*(b) \geq V_k^*(b)$. Let strategy σ_1 of player 1 in G_k plays uniformly for first k stages. Then $V_k^*(b) = \mathrm{val}^{\sigma_1^{k*}}(b) \geq \mathrm{val}^{\sigma_1}(b) = \mathrm{val}^{\sigma_{\text{unif}}}(b)$. □

Theorem 1. *Let $k \geq 1$. Then $V^*(b_{\text{init}}) - V_k^*(b_{\text{init}}) \leq -(1 - p_k) \min_b \mathrm{val}^{\sigma_{\text{unif}}}(b)$ for $p_k = [k\overline{R} - \mathrm{val}^{\sigma_{\text{unif}}}(b_{\text{init}})]/k\overline{R}$.*

Proof. Let us define a game G_{k+} as a game with payoff $U_b^{k+}(\sigma_1,\sigma_2) = \mathbb{E}_{b,\sigma_1,\sigma_2}[\sum_{t=1}^{k} R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})]$ (i.e., total sum of rewards in first k stages) and let V_{k+}^* denote value function of G_{k+}. Since $R(\cdot) \leq 0$, we have $U_b^{\infty}(\sigma_1,\sigma_2) \leq U_b^{k+}(\sigma_1,\sigma_2)$ and $V_k^*(b) \leq V^*(b) \leq V_{k+}^*(b)$ (the first inequality follows from Proposition 1). We prove the statement by deriving bound on $V_{k+}^*(b_{\text{init}}) - V_k^*(b_{\text{init}})$.

Take (σ_1^-, σ_2^-) and (σ_1^+, σ_2^+) as the Nash equilibrium strategies in G_k and G_{k+}, respectively. Since both games G_k and G_{k+} correspond to the game G being played over k stages, the strategies can be used interchangeably and we get

$$\begin{aligned} U^k_{b_{\text{init}}}(\sigma_1^+,\sigma_2^-) \leq U^k_{b_{\text{init}}}(\sigma_1^-,\sigma_2^-) = V_k^*(b_{\text{init}}) \leq \\ \leq V_{k+}^*(b_{\text{init}}) = U^{k+}_{b_{\text{init}}}(\sigma_1^+,\sigma_2^+) \leq U^{k+}_{b_{\text{init}}}(\sigma_1^+,\sigma_2^-) \ . \end{aligned} \tag{6}$$

We now claim that the strategy profile (σ_1^+, σ_2^-) reaches the goal state with probability at least $p = [k\overline{R} - \mathrm{val}^{\sigma_{\text{unif}}}]/k\overline{R}$ in the first k stages. To prove this claim, we use the following facts:

1. We have that the payoff $\mathrm{val}^{\sigma_{\text{unif}}}(b_{\text{init}}) \leq U^{k+}_{b_{\text{init}}}$. This is based on two inequalities: $V_k^*(b_{\text{init}}) \leq U^{k+}_{b_{\text{init}}}$ (see Equation (6)) and $\mathrm{val}^{\sigma_{\text{unif}}}(b_{\text{init}}) \leq V_k^*(b_{\text{init}})$ (see Proposition 1).
2. Player 1 accumulates negative reward $\overline{R} = \max R(\cdot) < 0$ in every stage of the game (recall that all rewards are negative). Hence, in the case a goal state is not reached within k stages considered in G_{k+}, he accumulates a negative payoff $k\overline{R}$.
3. Assuming that (σ_1^+, σ_2^-) reaches a goal state within k stages with probability p, the total accumulated negative payoff of (σ_1^+, σ_2^-) is $U^{k+}_{b_{\text{init}}}(\sigma_1^+, \sigma_2^-) \leq (1-p)k\overline{R}$.

From (2) we know that $\mathrm{val}^{\sigma_{\text{unif}}} \leq (1-p)k\overline{R}$. Solving this inequality for p gives us $p \geq [k\overline{R} - \mathrm{val}^{\sigma_{\text{unif}}}]/k\overline{R}$.

By definition $U^{k+}_{b_{\text{init}}}(\sigma_1^+, \sigma_2^-) - U^{k}_{b_{\text{init}}}(\sigma_1^+, \sigma_2^-) = \mathbb{E}_{b_{\text{init}},\sigma_1^+,\sigma_2^-}[-\mathrm{val}^{\sigma_{\text{unif}}}(b^{(k+1)})]$. The game reaches S_G with probability p (in which case $\mathrm{val}^{\sigma_{\text{unif}}}(b^{(k+1)}) = 0$), hence we have $\mathbb{E}_{b_{\text{init}},\sigma_1^+,\sigma_2^-}[-\mathrm{val}^{\sigma_{\text{unif}}}(b^{(k+1)})] \leq -(1-p) \min_b \mathrm{val}^{\sigma_{\text{unif}}}(b)$. As $V^*(b_{\text{init}}) - V_k^*(b_{\text{init}}) \leq U^{k+}_{b_{\text{init}}}(\sigma_1^+, \sigma_2^-) - U^{k}_{b_{\text{init}}}(\sigma_1^+, \sigma_2^-)$ we get the proof. □

Observe that p_k in Theorem 1 is monotonically increasing in k with limit $\lim_{k\to\infty} p_k = 1$. Hence we can find k such that $-(1-p_k) \min_b \mathrm{val}^{\sigma_{\text{unif}}}(b) \leq \varepsilon$.

Corollary 1. *Let $\varepsilon > 0$. Then for $k \geq K = -[\min_b \mathrm{val}^{\sigma_{\text{unif}}}(b) \cdot \mathrm{val}^{\sigma_{\text{unif}}}(b_{\text{init}})]/\varepsilon\overline{R}$, we have $V^*(b_{\text{init}}) - V_k^*(b_{\text{init}}) \leq \varepsilon$.*

Proof. By Theorem 1, for $k \geq 1$, we have $V^*(b_{\text{init}}) - V_k^*(b_{\text{init}}) \leq -[\min_b \mathrm{val}^{\sigma_{\text{unif}}}(b) \cdot \mathrm{val}^{\sigma_{\text{unif}}}(b_{\text{init}})]/k\overline{R}$. Hence, for $k \geq K$, we get $V^*(b_{\text{init}}) - V_k^*(b_{\text{init}}) \leq \varepsilon$. □

3.2 Finite-Horizon OS-POSGs

The k-cutoff game introduced in Section 3.1 corresponds to a finite-horizon OS-POSG with total-sum objective. While the prior results for OS-POSGs consider only infinite-horizon problems with discounted-sum objective, a straightforward reduction can be used to show that most of the results can be carried on towards setting with finite-horizon H and total sum of rewards $\mathbb{E}[\sum_{t=1}^{H} R(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})]$ as the objective. Consider a OS-POSG game structure $G = \langle S, A_1, A_2, O, T, R, b_{\text{init}} \rangle$ with finite-horizon H and total sum objective. Let us define an equivalent infinite-horizon discounted-sum game $G^\gamma = \langle S^\gamma, A_1, A_2, O, T^\gamma, R^\gamma, b_{\text{init}} \rangle$ by expanding the state space of G:

- $S^\gamma = \{s_h \mid s \in S, 0 \leq h \leq H\}$ (i.e., s_h corresponds to a state s in G when h stages remain to be played),
- $T^\gamma(o, s'_{h-1} \mid s_h, a_1, a_2) = T(o, s' \mid s, a_1, a_2)$ for $h \geq 1$ (i.e., after each transition there is one less stage to be played in the remainder of the game),
- $T^\gamma(\tilde{o}, s_0 \mid s_0, a_1, a_2) = 1$ for an arbitrary fixed observation $\tilde{o} \in O$ (i.e., when 0 stages remain transitions in the game stop),
- $R^\gamma(s_h, a_1, a_2) = R(s, a_1, a_2)/\gamma^{H-h}$ for $h \geq 1$ and $R^\gamma(s_0, a_1, a_2) = 0$.

Observe that the definition of rewards R^γ ensures that no reward is allocated after H stages of the game pass (since a state s_0 is reached). Moreover the discounting terms in the discounted-sum objective $\mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} R^\gamma(s^{(t-1)}, a_1^{(t)}, a_2^{(t)})]$ cancel out and hence the objective of G^γ coincide with the total-sum objective of G.

Applying the HSVI algorithm to the OS-POSG G^γ is, however, impractical as the value of state s_h does not correspond to the total-sum payoff P1 achieves in the remaining h stages. Due to the multiplication of the rewards by $1/\gamma^{H-h}$, the value of s_h is also multiplied by $1/\gamma^{H-h}$. We show that we can avoid this issue and solve G^γ for $\gamma = 1$ (referred to as $G^{\gamma=1}$) by altering the sequence ρ used in the HSVI algorithm (see Equation (2))—we term this algorithm FH-HSVI.

For simplicity of discussion, let us partition the optimal value function V^* of $G^{\gamma=1}$ based on the number of remaining stages in the game

$$V_h^*(b) = V^*(b|_h) \qquad (7)$$
$$\text{for } b \in \Delta(S), b|_h \in \Delta(S^\gamma) : b|_h(s_h) = b(s)\ .$$

This allows us to rewrite Equation (1) as

$$V_h^*(b) = [HV_{h-1}^*](b) = \max_{\pi_1} \min_{\pi_2} \Big[\mathbb{E}_{b,\pi_1,\pi_2}[R(s, a_1, a_2)] + \\ + \sum_{a_1,o} \mathbb{P}_{b,\pi_1,\pi_2}[a_1, o] \cdot V_{h-1}^*(\tau(b, a_1, \pi_2, o)) \Big] \qquad (8)$$

and obtain a Bellman equation for $G^{\gamma=1}$ in a traditional form for finite-horizon problems.

Observe that it clearly holds that $V_0^*(b) = 0$ for every $b \in \Delta(S)$ as no rewards are allocated after H stages when the game reaches a state s_0. We can therefore set the initial lower and upper bounds $\underline{V}_0$ and $\overline{V}_0$ on V_0^* in the HSVI algorithm to 0 as well. This means that the Explore procedure eventually reaches a belief $b|_0$ where the gap $\overline{V}_0(b) - \underline{V}_0(b) = 0$ (unless it is terminated before reaching H-th stage). Unlike in the discounted setting, the sequence of desired gaps ρ hence need not be strictly increasing. Instead, we let $\rho(t) = \epsilon - (t-1)\epsilon/H$ (i.e., $\rho(H+1) = 0$ after H-th stage is considered by Explore which is sufficient for the termination of the trial as the gap $\overline{V}_0(b) - \underline{V}_0(b) = 0$).

Importantly, since the payoff of any strategy in H'-horizon game can be bounded by $[H' \min R(\cdot), H' \max R(\cdot)]$, a value function $V_{H'}^*$ is $\delta_{H'}$-Lipschitz continuous for $\delta_{H'} = H'[\max R(\cdot) - \min R(\cdot)]/2$ (cf. [Horák *et al.*, 2017, Lemma 4]). Similarly to the discounted case, this allows us to consider only $\delta_{H'}$-Lipschitz continuous lower and upper bounds $\underline{V}_{H'}$ and $\overline{V}_{H'}$ on $V_{H'}^*$. As a consequence we can verify that the convergence properties discussed in Section 2.1 hold for FH-HSVI:

(1) Each trial terminates for $t \leq H+1$. After H-th recursion step, the game is in one of the states with zero remaining time and $\overline{V}_0(b) - \underline{V}_0(b) = 0 \leq \rho(H+1)$.

(2) Point based update in $b^{(t-1)}$ ensures that all beliefs within hypersphere with radius $\varepsilon/2H\delta_H$ centered in $b^{(t-1)}$ have negative excess gap (Proposition 2).

(3) By the same packing argument, we can see that the hyperspheres induced by the beliefs with negative excess gap eventually cover the entire belief space (unless we achieve convergence before that).

Proposition 2. *Let* $b^{(t-1)} \in \Delta(S)$ *and* $\pi_1^{(t)}$*,* $\pi_2^{(t)}$ *be Nash equilibrium strategies of P1 and P2 in* $[H\overline{V}](b^{(t-1)})$ *and* $[H\underline{V}](b^{(t-1)})$*, respectively. Assume that all beliefs* $\tau(b^{(t-1)}, a_1, \pi_2^{(t)}, o)$ *reachable when following* $(\pi_1^{(t)}, \pi_2^{(t)})$ *have negative excess gap* $\text{excess}_t(\tau(b^{(t-1)}, a_1, \pi_2^{(t)}, o)) \leq 0$. *Then after performing the point-based update in* $b^{(t-1)}$ *we have* $\text{excess}_{t-1}(b') \leq 0$ *for every* b' *such that* $\|b^{(t-1)} - b'\| \leq \varepsilon/2H\delta_H$.

Proof. Let $u_{V,b}(\pi_1, \pi_2)$ denote the utility in stage game $[HV](b)$ when the players use stage strategies (π_1, π_2). Let $(\underline{\pi}_1, \pi_2^{(t)})$ and $(\pi_1^{(t)}, \overline{\pi}_2)$ be Nash equilibrium strategies in stage games $[H\underline{V}_{H-t}](b^{(t-1)})$ and $[H\overline{V}_{H-t}](b^{(t-1)})$, respectively. By similar argument to the proof of [Horák *et al.*, 2017, Lemma 4], we have

$$\begin{aligned}
&[H\overline{V}_{H-t}](b^{(t-1)}) - [H\underline{V}_{H-t}](b^{(t-1)}) = \\
&\quad = u_{\overline{V}_{H-t}, b^{(t-1)}}(\pi_1^{(t)}, \overline{\pi}_2) - u_{\underline{V}_{H-t}, b^{(t-1)}}(\underline{\pi}_1, \pi_2^{(t)}) \\
&\quad \leq u_{\overline{V}_{H-t}, b^{(t-1)}}(\pi_1^{(t)}, \pi_2^{(t)}) - u_{\underline{V}_{H-t}, b^{(t-1)}}(\pi_1^{(t)}, \pi_2^{(t)}) \\
&\quad = \sum_{a_1,o} \mathbb{P}_{b^{(t-1)}, \pi_1^{(t)}, \pi_2^{(t)}}[a_1, o] \big[\overline{V}_{H-t}(\tau(b^{(t-1)}, a_1, \pi_2^{(t)}, o)) - \\
&\qquad\qquad \underline{V}_{H-t}(\tau(b^{(t-1)}, a_1, \pi_2^{(t)}, o)) \big] \\
&\quad \leq \rho(t) = \rho(t-1) - \varepsilon/H\ .
\end{aligned}$$

The point-based update in $b^{(t-1)}$ hence ensures that $\overline{V}_{H-t+1}(b^{t-1}) - \underline{V}_{H-t+1}(b^{t-1}) \leq \rho(t-1) - \varepsilon/H$. Since $\underline{V}_{H-t+1}$ and $\overline{V}_{H-t+1}$ are δ_H-Lipschitz continuous (observe that $\delta_H \geq \delta_{H'}$ for every $0 \leq H' \leq H$), we have that $\overline{V}_{H-t+1}(b^{t-1}) - \underline{V}_{H-t+1}(b^{t-1}) \leq \rho(t-1)$ for every belief b' such that $\|b' - b^{(t-1)}\| \leq \varepsilon/2H\delta_H$. □

Non-zero termination utilities The definition of k-cutoff game assumes that a one-time payoff $\text{val}^{\sigma_{\text{unif}}}(b^{(k)})$ is allocated after k stages of the game are played depending on the belief $b^{(k)}$ at that time. This means that k-cutoff game is technically not a finite-horizon OS-POSG where no rewards are assigned after $H = k$ stages. However, we argue that the same algorithm can be applied upon setting $V_0^*(b) = \underline{V}_0(b) = \overline{V}_0(b) = \text{val}^{\sigma_{\text{unif}}}(b)$ since

- In the worst case, the `Explore` procedure reaches a belief $b|_0$ where $\overline{V}_0(b) - \underline{V}_0(b) = 0$,
- The payoff of any strategy in H'-horizon game is bounded by $[\min_b V_0^*(b) + H' \min R(\cdot), \max_b V_0^*(b) + H' \max R(\cdot)]$, and hence $V_{H'}^*$ is Lipschitz continuous.

3.3 Solving POSSPGs

Due to Corollary 1, we can use a k-cutoff games for $k \geq K$ to approximate the value of POSSPG within ε accuracy. The theoretical value of K, however, could be impractically large. For practical use, we suggest keeping track of an auxiliary upper bound $\overline{V}^+$ on the value of the POSSPG V^* throughout the course of the algorithm. This allows us to terminate as soon as $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}}) \leq \epsilon$ (which can be much earlier before we reach K).

The auxiliary upper bound $\overline{V}^+(b_{\text{init}})$ is described as follows. Let G_{k+} be a variant of k-cutoff game in which the player 1 is perfectly informed and if the goal state is not reached in first k stages, then the reward for player 1 is 0 instead of the continuation reward $\text{val}^{\sigma_{\text{unif}}}(b^{(k)})$. Let V_{k+}^* denote value function of G_{k+}. The function $\overline{V}^+$ is initialized as V_{0+}^*. While the procedure `solvecutoff(k)` is called, every time the point-based updates are performed on $\overline{V}^k$ and $\underline{V}^k$ at belief b, then point-based update is also performed on $\overline{V}^+$ at belief b. As $V_{0+}^*(b_{\text{init}}) = 0$ and $V^*(b_{\text{init}}) < 0$, the initial value of $\overline{V}^+(b_{\text{init}})$ is indeed an upper bound on $V^*(b_{\text{init}})$. The point based updates ensure that $\overline{V}^+(b_{\text{init}})$ remains an upper bound on $V^*(b_{\text{init}})$.

Our algorithm (Algorithm 2) starts by setting $k = 1$ and initializing $\underline{V}^k$, $\overline{V}^k$, and $\overline{V}^+$. For each value of k, we perform the following operations.

1. Solve k-cutoff game G_k using FH-HSVI from Section 3.2. This gives a lower bound $\underline{V}^k(b_{\text{init}})$ on $V_k^*(b_{\text{init}})$ and consequently a lower bound on $V^*(b_{\text{init}})$.[4] The gap $\overline{V}^k(b_{\text{init}}) - \underline{V}^k(b_{\text{init}})$ converges faster than the gap $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}})$ used for the algorithm termination. Therefore when solving G_k, we aim for a gap that the is tighter than the target gap ε.[5] This ensures that the termination gap $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}})$ can reach the desired value and the algorithm can terminate.
2. For every point-based update on $\overline{V}^k$ and $\underline{V}^k$ at belief b (line 17 of Algorithm 2), perform a point-based update on $\overline{V}^+$ at belief b. (line 18 of Algorithm 2)
3. If the gap $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}}) < \varepsilon$ or if $k > K$, then algorithm is terminated. Otherwise increment k by 1.

Algorithm 2: `HSVI` algorithm for `POSSPGs`

```
1  $k \leftarrow 1$
2  $\underline{V}^k \leftarrow \text{val}^{\sigma_{\text{unif}}}$
3  $\overline{V}^+ \leftarrow V_{0+}^*$
4  while $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}}) > \varepsilon$ and $k \leq K$ do
5      $\overline{V}^k \leftarrow V_{0+}^*$
6      solvecutoff($k$)
7      $k \leftarrow k + 1$

8  procedure solvecutoff($k$)
9      while $\overline{V}^k(b_{\text{init}}) - \underline{V}^k(b_{\text{init}}) > \eta * \varepsilon$ and
         $\overline{V}^+(b_{\text{init}}) - \underline{V}^k(b_{\text{init}}) > \varepsilon$ do
10         Explore($b_{\text{init}}$, 1)

11 procedure Explore($b^{(t)}$, $t$)
12     if $\text{excess}_t(b^{(t)}) \leq 0$ then return
13     $\pi_1^{(t)} \leftarrow$ optimal strategy of P1 in $[H\overline{V}^{k-t}](b^{(t)})$
14     $\pi_2^{(t)} \leftarrow$ optimal strategy of P2 in $[H\underline{V}^{k-t}](b^{(t)})$
15     $b^{(t+1)} \leftarrow \arg\max_{a_1,o} \mathbb{P}_{b^{(t)},\pi_1^{(t)},\pi_2^{(t)}}[a_1, o] \cdot$
         $\text{excess}_{t+1}(\tau(b^{(t)}, a_1, \pi_2^{(t)}, o))$
16     Explore($b^{(t+1)}$, $t + 1$)
17     Perform point-based update of $\underline{V}^{k-t+1}$ and $\overline{V}^{k-t+1}$ in
         $b^{(t)}$
18     Perform point-based update of $\overline{V}^+$ in $b^{(t)}$
```

Final improvements We now suggest some modifications that we use in our implementation, which decrease the runtime of the algorithm. Instead of initializing k-cutoff game upper bound $\overline{V}^k$ (line 5) and auxiliary upper bound $\overline{V}^+$ (line 3) as $V_{0+}^* = 0$ for $k > 1$, we initialize them using points representing $\overline{V}^+$ in $(k-1)$-cutoff game. This initialization is indeed an upper bound on V^* since adding an additional round for which the game is not played as perfect information version of POSSPG can only result in the same or worse utility of player 1 in k-cutoff game compared to $(k-1)$-cutoff game. To avoid increased memory complexity of the algorithm, we keep only points that are not dominated when moving from k-cutoff game to $(k+1)$-cutoff game.

As V_k^* is a lower bound of V_{k+1}^*, the lower bound $\underline{V}^k$ of V_k^* (generated in line 6) is also a lower bound of V_{k+1}^*. So while performing `solvecutoff(k+1)` (line 6), we use $\underline{V}^k$ to initialize $\underline{V}^{k+1}$, instead of initializing as the value of uniform strategy. As $\underline{V}^k$ is greater than the value of uniform strategy, this initialization decreases the runtime of the algorithm.

Finally, the target value $\eta * \varepsilon$ of the gap $\overline{V}^k(b_{\text{init}}) - \underline{V}^k(b_{\text{init}})$ has effect on the runtime too. In our case, we observed the best results with the parameter $\eta = 0.9$.

4 Experiments

In this section, we present an experimental evaluation of the proposed algorithm (Algorithm 2) on the domain of pursuit-evasion games and show how it performs compared to traditional solution approaches with discount factor $\gamma < 1$.

4.1 Experiment Settings

In pursuit-evasion games ([Chung *et al.*, 2011; Isler and Karnad, 2008]), a team of K centrally controlled pursuers (we

[4] Note that the course of the future exploration (line 16) solely relies on strategies computed for k-cutoff game (lines 13 and 14).

[5] The amount of tightness is controlled by the parameter η, $0 < \eta \leq 1$ (line 9 of Algorithm 2).

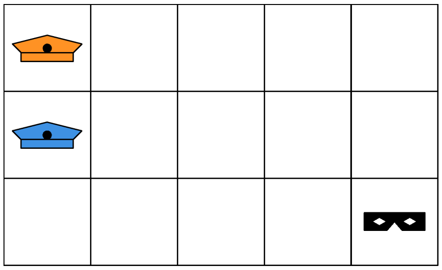

Figure 1: Example of the pursuit-evasion game.

consider a team of $K = 2$) is trying to locate and capture the evader — who is trying to avoid getting captured. The game is played on a grid (dimensions $3 \times N$), with the pursuers starting in the top-left corner and the evader in the bottom-right corner – see Figure 1. In each step, the units move to one of their adjacent locations (i.e., the actions of the evader are $A_2 = \{\text{left}, \text{right}, \text{up}, \text{down}\}$, while the actions available to the team of pursuers are joint actions for all units in the team, $A_1 = (A_2)^K$). The game ends when one of the units from the team of pursuers enters the same cell as the evader or directly swaps position with the evader. The reward for all transitions in the game is -1. The pursuer knows the location of their units, but the current location of the evader is not known.

All computational results have been obtained on computers equipped with *Intel Xeon Scalable Gold 6146* processors while limiting the runtime to 10 hours and RAM to 128 GB. We used CPLEX 12.9 to solve linear programs.

All solution methods were required to find an ϵ-optimal solution where ϵ was set to 1. Since the reward for all transitions in the game is -1, such setting allows us to find an optimal solution ± 1 move.

4.2 Algorithm Scalability

We compare the proposed algorithm (HSVI for POSSPGs) with the original HSVI algorithm for *discounted* OS-POSGs. Recall that the original algorithm [Horák *et al.*, 2017] is capable of solving games with discount factor $\gamma < 1$, and the obtained solution can thus be considered only as an approximation. Since the approximation quality is directly related to the discount factor (the undiscounted setting can be seen as a limit solution as $\gamma \to 1$), we use several values of discount factor γ, namely $\gamma = 0.95$, $\gamma = 0.99$, and $\gamma = 0.999$.

We report the results in Table 1. Observe that our algorithm is significantly faster than the original HSVI algorithm for solving the discounted approximation of the game with discount factor $\gamma = 0.999$. The computational time required by our algorithm is 25-37x smaller compared to the prior approach. If we further sacrifice the accuracy of the discounted approximation and further reduce the discount factor γ, we have to expect improvements in computational time when solving the discounted approximation of the game. The setting with $\gamma = 0.95$ is solved substantially faster compared to our approach with $\gamma = 1$. However, we have to expect significant degradation in solution quality as the contribution of future rewards diminishes quickly with $\gamma = 0.95$.

Instance	Our	Original HSVI algorithm		
($3 \times N$)	approach	$\gamma = 0.95$	$\gamma = 0.99$	$\gamma = 0.999$
3×3	2 s	1 s	4 s	58.5 s
3×4	54.5 s	4 s	133.5 s	2 032 s
3×5	607 s	7.5 s	1 385.5 s	15 259 s

Table 1: Scalability in the size of grid dimension N.

The computational benefits of our approach compared to using original HSVI algorithm to solve a discounted approximation with $\gamma = 0.999$ are further highlighted when solving the 3×6 instance. While our approach is able to approximate the value of the game within 1.47 precision after 10 hours (measured as the gap between upper and lower bounds computed by the algorithm), the gap for the original approach is 4.956.

5 Conclusion

We introduce a new algorithm for solving two-player zero-sum partially observable stochastic shortest-path games (POSSPGs) – a variant of a partially observable stochastic game with the undiscounted sum of rewards as an objective. We assume that the adversary has the perfect information, and thus our algorithm allows the agent to find robust strategies. We provide theoretical guarantees for the convergence of our algorithm and compare the performance with the algorithm for the discounted case. The results show that it is not feasible to use very large discount factors to approximate the total reward objective, and thus our novel algorithm applies in all scenarios where the future rewards should not be discounted.

Our algorithm is the first one to solve the class of partially observable SSPGs, and thus follow-up research focused on the scalability improvements would open possibilities of various applications in robotics or, for example, network security. As a second direction for future work, we intend to analyze reachability/safety objectives where the probability of reaching some of the target states is optimized.

Acknowledgements

This research was supported by the Czech Science Foundation (no. 19-24384Y), by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16 019/0000765 “Research Center for Informatics”, by the ERC CoG 863818 (ForM-SMArt), and by the Combat Capabilities Development Command 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 Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation here on.

References

[Bertsekas and Tsitsiklis, 1991] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[Bertsekas, 1995] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[Chen *et al.*, 2007] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin. Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach. *IEEE Transactions on Signal Processing*, 55(5):2294–2309, 2007.

[Chen *et al.*, 2020] Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Minimax Regret for Stochastic Shortest Path with Adversarial Costs and Known Transition. *arXiv preprint arXiv:2012.04053*, 2020.

[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.

[Delamer *et al.*, 2019] Jean Alexis Delamer, Yoko Watanabe, and Caroline P. Carvalho Chanel. Solving path planning problems in urban environments based on a priori sensor availability and execution error propagation. *AIAA Scitech 2019 Forum*, 2019.

[Egorov *et al.*, 2016] Maxim Egorov, Mykel J. Kochenderfer, and Jaak J. Uudmae. Target surveillance in adversarial environments using POMDPs. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 2473–2479, 2016.

[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*, pages 558–564, 2017.

[Horák *et al.*, 2018] Karel Horák, Branislav Bošanský, and Krishnendu Chatterjee. Goal-HSVI: Heuristic search value iteration for goal-POMDPs. *IJCAI International Joint Conference on Artificial Intelligence*, pages 4764–4770, 2018.

[Horák *et al.*, 2020] Karel Horák, Branislav Bošanský, Vojtěch Kovařík, and Christopher Kiekintveld. Solving zero-sum one-sided partially observable stochastic games. *arXiv preprint arXiv:2010.11243*, 2020.

[Isler and Karnad, 2008] Volkan Isler and Nikhil Karnad. The role of information in the cop-robber game. *Theoretical Computer Science*, 399(3):179–190, 2008.

[Lim *et al.*, 2013] Sejoon Lim, Christian Sommer, Evdokia Nikolova, and Daniela Rus. Practical route planning under delay uncertainty: Stochastic shortest path queries. In *Robotics: Science and Systems*, volume 8, pages 249–256, 2013.

[Neu *et al.*, 2012] Gergely Neu, Andras Gyorgy, and Csaba Szepesvári. The adversarial stochastic shortest path problem with unknown transition probabilities. In *Artificial Intelligence and Statistics*, pages 805–813, 2012.

[Norman *et al.*, 2005] Gethin Norman, David Parker, Marta Kwiatkowska, Sandeep Shukla, and Rajesh Gupta. Using probabilistic model checking for dynamic power management. *Formal aspects of computing*, 17(2):160–176, 2005.

[Patek and Bertsekas, 1999] Stephen D. Patek and Dimitri P. Bertsekas. Stochastic shortest path games. *SIAM Journal on Control and Optimization*, 37(3):804–824, 1999.

[Patek, 1999] Stephen D Patek. On partially observed stochastic shortest path problems. *Systems Engineering*, (i):1–14, 1999.

[Rosenberg and Mansour, 2020] Aviv Rosenberg and Yishay Mansour. Stochastic Shortest Path with Adversarially Changing Costs. *arXiv preprint arXiv:2006.11561*, 2020.

[Saisubramanian *et al.*, 2019] S. Saisubramanian, K. H. Wray, L. Pineda, and S. Zilberstein. Planning in stochastic environments with goal uncertainty. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1649–1654, 2019.

[Smith and Simmons, 2004] Trey Smith and Reid Simmons. Heuristic search value iteration for POMDPs. In *20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 520–527, 2004.

[Smith and Simmons, 2005] Trey Smith and Reid Simmons. Point-based POMDP algorithms: improved analysis and implementation. In *21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 542–549, 2005.