HSVI-based Online Minimax Strategies for Partially Observable Stochastic Games with Neural Perception Mechanisms

Rui Yan

RUI.YAN@CS.OX.AC.UK

Gabriel Santos

GABRIEL.SANTOS@CS.OX.AC.UK

Gethin Norman

GETHIN.NORMAN@GLASGOW.AC.UK

David Parker

DAVID.PARKER@CS.OX.AC.UK

Marta Kwiatkowska

MARTA.KWIATKOWSKA@CS.OX.AC.UK

1Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK

2School of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK

Abstract

We consider a variant of continuous-state partially-observable stochastic games with neural perception mechanisms and an asymmetric information structure. One agent has partial information, with the observation function implemented as a neural network, while the other agent is assumed to have full knowledge of the state. We present, for the first time, an efficient online method to compute an $\varepsilon$-minimax strategy profile, which requires only one linear program to be solved for each agent at every stage, instead of a complex estimation of opponent counterfactual values. For the partially-informed agent, we propose a continual resolving approach which uses lower bounds, pre-computed offline with heuristic search value iteration (HSVI), instead of opponent counterfactual values. This inherits the soundness of continual resolving at the cost of pre-computing the bound. For the fully-informed agent, we propose an inferred-belief strategy, where the agent maintains an inferred belief about the belief of the partially-informed agent based on (offline) upper bounds from HSVI, guaranteeing $\varepsilon$-distance to the value of the game at the initial belief known to both agents.

Keywords: Minimax strategies, continual resolving, partially observable stochastic games.

1. Introduction

Partially-observable stochastic games (POSGs) are a modelling formalism that enables strategic reasoning and (near-)optimal synthesis of strategies and equilibria in multi-agent settings with partial observations and uncertainty. One-sided POSGs (Horák et al., 2023) are a tractable subclass of two-agent, zero-sum POSGs with an asymmetric information structure, where only one agent has partial information while the other agent is assumed to have full knowledge. This is well suited to autonomous safety- or security-critical settings, such as patrolling or pursuit-evasion games, which require reasoning about worst-case assumptions. Since real-world settings increasingly often utilise neural networks (NNs) for perception tasks such as localisation and object detection, one-sided neuro-symbolic POSGs (one-sided NS-POSGs) were introduced (Yan et al., 2023). In this model the agent with partial information observes the environment only through a trained NN classifier, and consequently the game is generalised to continuous environments, to align with NN semantics, while observations remain discrete. A point-based NS-HSVI method was developed to approximate values of one-sided NS-POSGs working with (polyhedral decompositions of) the continuous space.

Strategy synthesis for continuous games is more challenging than for the finite-state case (Horák et al., 2023), since continuous-state spaces lead to an infinite number of strategies and discretisation suffers from the curse of dimensionality. Several offline methods exist, based on counterfactual
regret minimisation, heuristics or reinforcement learning (see Related Work, below). In this paper, we consider online methods, which can improve efficiency and adaptability. The best performing online method (Moravčík et al., 2017) continually resolves a local strategy that only keeps track of the agent’s belief of its opponent state and a vector of opponent counterfactual values. Apart from Horák et al. (2023), existing continual resolving approaches (Moravčík et al., 2017; Šustr et al., 2019; Schmid et al., 2023) are for extensive form games (EFGs), and cannot be directly applied to POSGs. This is because, although the two formalisms are connected (Kovařík et al., 2022), transitioning between them is not straightforward.

Contributions. We develop a continual resolving approach for one-sided NS-POSGs, addressing several challenges. Firstly, existing continual resolving approaches need to estimate the opponent’s counterfactual values by solving a subgame at each stage (Moravčík et al., 2017), which would be intractable for continuous games. Instead, for the agent with partial observation (Ag₁), we use the lower bound computed offline by NS-HSVI (Yan et al., 2023), giving a polyhedral bound without solving a subgame. At each stage, we solve a linear program (LP), whose size is linear in the number of states in the current belief rather than the number of states reached, to compute the agent’s action choice and update the lower bound. Thus, a stage strategy is computed online, in the spirit of continual resolving, for each situation as it arises during execution, instead of storing a complete strategy. Although we require offline computation for NS-HSVI, existing continual resolving approaches need to train deep counterfactual value networks to solve the subgame. Importantly, our NS-HSVI continual resolving does not lose soundness, i.e., ε-exploitability (Burch et al., 2014).

We can use any synthesis method for fully-observable stochastic games to generate an ε-minimax strategy for the fully-informed agent (Ag₂) (Yan et al., 2022). However, solving the fully-observable case would generate a complete strategy, which can be costly in terms of memory. We instead propose an online inferred-belief strategy by observing that, by using the offline upper bound from NS-HSVI, Ag₂ only needs to keep track of an inferred belief about Ag₁’s belief and solve an LP, linear in the number of states in the current belief, to synthesise an action and the next inferred-belief of Ag₂. Since Ag₂ is fully informed, it does not need to store its belief. This allows us to generate a simpler strategy than the complete strategy for Ag₂, which guarantees the value at the initial belief, known to both agents, but cannot optimally employ the suboptimal actions of Ag₁ during play.

Summarising the contribution, we present, for the first time, an efficient online method to compute an ε-minimax strategy profile for one-sided NS-POSGs, a variant of two-player continuous-state POSGs with neural perception mechanisms, by exploiting bounds pre-computed offline by a variant of HSVI. We implement our approach, evaluate it on a pursuit-evasion model inspired by mobile robotics and investigate the synthesised agent strategies.

Related Work. Existing offline strategy synthesis methods for one-sided POSGs include a space partition approach (Zheng et al., 2022), a point-based approximate algorithm for continuous observations (Zheng et al., 2023), projection to POMDPs based on factored representations (Carr et al., 2021) and HSVI algorithms for finite (Horák et al., 2023) and continuous-state spaces (Yan et al., 2023). Since POSGs and EFGs are connected through factored-observation stochastic games (Kovařík et al., 2022), we next review relevant methods for two-agent zero-sum EFGs.

Counterfactual Regret Minimization (CFR) (Zinkevich et al., 2007) exploits the fact that the time-averaged strategy profile of regret minimizing algorithms converges to an ε-minimax strategy profile, in two-agent zero-sum EFGs with imperfect information. Since its introduction, a variety of CFR variants have been proposed and successfully applied to games (Lanctot et al., 2009; Lisý et al.,
2015; Burch et al., 2014). However, since the CFR-based approaches require iterative traversal of the game tree, they become intractable when the tree is large. Additionally, these approaches are offline algorithms, returning a complete solution strategy that is difficult to represent and store.

A number of algorithms based on game-theoretic learning models such as reinforcement learning and heuristic search have also been proposed, which are able to compute strategies for two-agent zero-sum games with imperfect information, including (Bosansky et al., 2014; Heinrich et al., 2015; Lanctot et al., 2017; McAleer et al., 2021) and heuristic search value iteration (HSVII) (Yan et al., 2023; Delage et al., 2023), which we utilise in our work. However, these approaches are also offline algorithms and unable to refine strategies at test time.

The most related approach is continual resolving used in Horák et al. (2023), which is based on DeepStack (Moravčík et al., 2017), although other variants have also been proposed, e.g., (Šustr et al., 2019; Schmid et al., 2023; Brown et al., 2020). Both Horák et al. (2023) and our work are on DeepStack (Moravčík et al., 2017), although other variants have also been proposed, e.g., (Šustr et al., 2019; Schmid et al., 2023; Brown et al., 2020). Both Horák et al. (2023) and our work are variants of continual resolving for one-sided POSGs, except we consider continuous-state spaces. Under the belief update in Horák et al. (2023), the current state could be missed and the belief might be empty. We assume a uniform stage strategy to fix this issue and ensure that the true state is always in the current belief. The LP size in Horák et al. (2023) is fixed, while the LP in our case varies at each stage because of no prior enumeration of all reachable states. Horák et al. (2023) performs the belief update before Ag taking action and observing, whereas we update the belief using the action and the next observation, which results in a more accurate belief.

2. Background

We briefly review the model of Yan et al. (2023), which generalises one-sided POSGs (Horák et al., 2023) to continuous-state spaces and allows neural perception mechanisms. Let \( \mathbb{P}(X) \) and \( \mathbb{F}(X) \) denote the spaces of probability measures and functions on a Borel space \( X \), respectively.

**One-sided NS-POSGs.** A one-sided neuro-symbolic POSG (NS-POSG) \( C \) is a two-player zero-sum infinite-horizon game with discrete actions and observations, where one player (\( A_1 \)) is partially informed and the other (\( A_2 \)) is fully informed. Unlike Horák et al. (2023), the game is played in a closed continuous environment \( S_E \), which \( A_1 \) perceives only using perception function \( \text{obs}_1 \) given as a (trained) ReLU NN classifier that maps environment states to so called percepts, ranging over a finite set \( \text{Per}_1 \). The use of classifiers is aligned with, e.g., object detection or vision tasks in autonomous systems. We further assume that \( A_1 \) has a discrete local state space \( \text{Loc}_1 \), which is observable to both agents, and that \( A_2 \) has full knowledge of the environment’s state.

A game \( C \) comprises agents \( A_1 = (S_1, A_1, \text{obs}_1, \delta_1), A_2 = (A_2) \) and environment \( E = (S_E, \delta_E) \), where \( S_1 = \text{Loc}_1 \times \text{Per}_1 \); \( A = A_1 \times A_2 \) are joint actions; \( \text{obs}_1 : (\text{Loc}_1 \times S_E) \rightarrow \text{Per}_1 \) is \( A_1 \)'s perception function (note that we allow NNs to additionally depend on local states); \( \delta_1 : (S_1 \times A) \rightarrow \mathbb{P}(\text{Loc}_1) \) is \( A_1 \)'s local transition function; and \( \delta_E : (\text{Loc}_1 \times S_E \times A) \rightarrow \mathbb{P}(S_E) \) is \( E \)'s finite-branching function. We work in the belief space \( S_B \subseteq \mathbb{P}(S) \), where \( S = S_1 \times S_E \), and assume an initial belief \( b^{\text{init}} \) using the particle-based representation (Porta et al., 2006; Doucet et al., 2001). A belief of \( A_1 \) is given by \( b = (s_1, b_1) \), where \( s_1 \in S_1, b_1 \in \mathbb{P}(S_E) \) and \( b_1 \) is represented by a weighted particle set \( \{(s_{E,i}^1, \kappa_i)\}_{i=1}^{N_1} \) where \( \kappa_i \geq 0 \) and \( \sum_{i=1}^{N_1} \kappa_i = 1. \)

The game starts in a state \( s = (s_1, s_E) \), where \( s_1 = (\text{loc}_1, \text{per}_1) \in S_1 \), and \( s \) is sampled from \( b^{\text{init}} \). At each stage of the game, both agents concurrently choose one of their actions. If \( a = (a_1, a_2) \in A \) is chosen, the local state \( \text{loc}_1 \) of \( A_1 \) is updated to \( \text{loc}'_1 \in \text{Loc}_1 \) via \( \delta_1(s_1, a) \), while the environment updates its state to \( s_{E}' \in S_E \) via \( \delta_E(\text{loc}_1, s_E, a) \). Finally, \( A_1 \), based on \( \text{loc}'_1 \), generates
its percep \( per_1' = \text{obs}_1(loc_1', s_E') \) at \( s_E' \) and \( C \) reaches the state \( s' = ((loc_1', per_1'), s_E') \). The probability of transitioning from \( s \) to \( s' \) under \( a \) is \( \delta(s, a)(s') = \delta_1(s_1, a)(loc_1')\delta_E(loc_1, s_E, a)(s_E') \).

**Strategies.** We distinguish between a *history* \( \pi \) (a sequence of states and joint actions, where \( \pi(k) \) is the \((k+1)\)th state, and \( \pi[k] \) is the \((k+1)\)th action) and a (local) *action-observation history* (AOH) for \( Ag_i \) (a sequence of its observations and actions). For the fully-informed \( Ag_2 \), an AOH is a history. A strategy of \( Ag_i \) is a mapping \( \sigma_i : FPaths_{C,i} \to \mathbb{P}(A_i) \), where \( FPaths_{C,i} \) is the set of \( Ag_i \)'s finite AOHs. A (strategy) profile \( \sigma = (\sigma_1, \sigma_2) \) is a pair of strategies and we denote by \( \Sigma_i \) and \( \Sigma \) the sets of strategies of \( Ag_i \) and profiles. The choices for the players after a history \( \pi \) are given by *stage strategies* for \( Ag_1 \) this is a distribution \( u_1 \in \mathbb{P}(A_1) \) and for \( Ag_2 \) a function \( u_2 : S \to \mathbb{P}(A_2) \), i.e., \( u_2 \in \mathbb{P}(A_2 \mid S) \). Given a belief \((s_1, b_1)\), if \( Ag_1 \) chooses \( a_1 \), assumes \( Ag_2 \) chooses \( u_2 \) and observes \( s_1' \), then the updated belief of \( Ag_1 \) via Bayesian inference is denoted \((s_1', b_1^{s_1,a_1,u_2,s_1'})\).

**Objectives and values.** We focus on infinite-horizon expected accumulated reward \( \mathbb{E}_b^\gamma[Y] \) when starting from \( b \) under \( \sigma \), where \( Y(\pi) = \sum_{k=0}^{\infty} \beta^k r(\pi(k), \pi[k]) \) for an infinite history \( \pi \), reward structure \( r : (S \times A) \to \mathbb{R} \) and discount \( \beta \in (0, 1) \), and \( Ag_1 \) and \( Ag_2 \) maximise and minimise the expected value. Given \( \epsilon \geq 0 \), a profile \( \sigma^* = (\sigma_1^*, \sigma_2^*) \) is an \( \epsilon \)-minimax strategy profile if for any \( b \in S_B \), \( \mathbb{E}^\gamma_{b}^\sigma[Y] \leq \mathbb{E}^\gamma_{b}^{\sigma_1^*, \sigma_2^*}[Y] + \epsilon \) for all \( \sigma_2 \) and \( \mathbb{E}^\gamma_{b}^\sigma[Y] \geq \mathbb{E}^\gamma_{b}^{\sigma_1, \sigma_2^*}[Y] - \epsilon \) for all \( \sigma_1 \). If \( \epsilon = 0 \), then \( \mathbb{E}^\gamma_{b}^\sigma[Y] \) is the value of \( C \), denoted \( V^* \).

**One-sided NS-HSVI.** HSVI is an anytime algorithm that approximates the value \( V^* \) via *lower* and *upper bound* functions, updated through heuristically generated beliefs. One-sided NS-HSVI (Yan et al., 2023) works with the continuous-state space of a one-sided NS-POSG using a generalisation of \( \alpha \)-functions, similar to Porta et al. (2006), except it uses *polyhedral* representations induced from NNs instead of Gaussian mixtures. For \( \epsilon > 0 \), one-sided NS-HSVI returns lower and upper bound functions \( V_{lb}^\gamma, V_{ub}^\gamma \in \mathbb{F}(S_B) \) to approximate \( V^* \) such that \( V_{lb}^\gamma(b) \leq V^*(b) \leq V_{ub}^\gamma(b) \) for all \( b \in S_B \) and \( V_{ub}^\gamma(b^{\text{init}}) - V_{lb}^\gamma(b^{\text{init}}) \leq \epsilon \). Given \( f : S \to \mathbb{R} \) and belief \((s_1, b_1)\), let \((f, (s_1, b_1)) = \int_{s_E \in S_E} f(s_1, s_E) b_1(s_E) ds_E \). The lower bound \( V_{lb}^\gamma \) is represented via a finite set \( \Gamma \subseteq \mathbb{F}(S) \) of *piecewise-constant* (PWC) \( \alpha \)-functions such that \( V_{lb}^\gamma(s_1, b_1) = \max_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1) \rangle \). The upper bound \( V_{ub}^\gamma \) is represented by a finite set of belief-value pairs \( \Upsilon \subseteq S_B \times \mathbb{R} \) and computed via an LP.

3. NS-HVSI Continual resolving

Continual resolving, e.g., (Moravčík et al., 2017), is an online method for computing an \( \epsilon \)-minimax strategy in two-player, zero-sum imperfect information EFGs; it keeps track of an agent’s belief of its opponent state and opponent counterfactual values to build and solve a subgame to synthesise choices, without building a complete strategy. It is *sound*, in computing an \( \epsilon \)-minimax strategy, but can be expensive as it needs to estimate opponent counterfactual values by traversing the game tree.

We now present a novel variant of continual resolving, which utilises the *lower bound* function \( V_{lb}^\gamma \) computed by one-sided NS-HSVI to synthesise an \( \epsilon \)-minimax strategy for \( Ag_1 \) that achieves the desired \( \epsilon \) distance to the value function at the initial belief. The method is efficient as it only requires solving a single LP at each stage. We first introduce the following minimax operator.

**Definition 1 (Minimax)** The minimax operator \( T : \mathbb{F}(S_B) \to \mathbb{F}(S_B) \) is given by:

\[
[T \sigma](s_1, b_1) = \max_{a_1 \in \Pi(A_1)} \min_{u_2 \in \Pi(A_2 | S)} \mathbb{E}_{(s_1, b_1), u_1, u_2} [r(s, a)] \\
+ \beta \sum_{(a_1, s_1') \in A_1 \times S_1} P(a_1, s_1' | (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1, a_1, u_2, s_1'})
\]
Compared with the LP in Yan et al. (2023) for solving their constraints, we can estimate the opponent counterfactual values for a belief and a PWC function by solving the LP presented below, thus avoiding the need to keep track of a belief and a PWC function. Motivated by Horák et al. (2023, Section 9.2), our online game-playing algorithm NS-HSVI continual resolving, see Algorithm 1, generates a strategy for $Ag_1$, denoted $\sigma_{init}^{lb}$, by using the HSVI lower bound instead of opponent counterfactual values used in Moravčík et al. (2017). Since we have pre-computed $V^T_{lb}$ offline, our NS-HSVI continual resolving only keeps track of a belief $(s_1, b_1)$ and a PWC function $\alpha_1$ in the the convex hull, Conv($\Gamma$), of $\Gamma$. Importantly, $\Gamma$ can be used to compute an action to play and update the tracking information at each stage (a belief and a PWC function) by solving the LP presented below, thus avoiding the need to estimate the opponent counterfactual values.

**Definition 2 (Stage strategy)** For $((s_1, b_1), \alpha_1) \in S_B \times \text{Conv}(\Gamma)$ where $b_1$ is represented by $\{ (s^i_E, \kappa_i) \}_{i=1}^{N_b}$, a stage strategy $u_{lb}^i$ for $Ag_1$ is such that $u_{lb}^i(a_1) = p^{*a_1}$ for $a_1 \in A_1$, where $(v^*_{s^i_E})_{i=1}^{N_b}$, $(\lambda^{a_1, s'_1})_{(a_1, s'_1) \in A_1 \times S_1, \alpha_1 \in \Gamma}$ and $(p^{*a_1})_{a_1 \in A_1}$ is a solution to the following LP:

\[
\begin{align*}
\text{maximise} \quad & \sum_{i=1}^{N_b} \kappa_i v_{s^i_E} \\
\text{subject to} \quad & v_{s^i_E} \leq \sum_{a_1 \in A_1} p^{a_1} r((s_1, s^i_E), (a_1, a_2)) + \beta \sum_{(a_1, s'_1) \in A_1 \times S_1} \alpha(a_1, s'_1) \alpha(s'_1, s^i_E) \lambda^{a_1, s'_1} - \delta((s_1, s^i_E), (a_1, a_2))(s'_1, s^i_E) \sum_{\alpha \in \Gamma} \lambda^{a_1, s'_1} \alpha(s'_1, s^i_E) \quad & 1 \leq i \leq N_b \text{ and } a_2 \in A_2 \\
& v_{s^i_E} \leq \alpha(s_1, s^i_E) \quad & 1 \leq i \leq N_b \\
& \lambda^{a_1, s'_1} \geq 0 \quad & a_1 \in A_1, \ s'_1 \in S_1 \text{ and } \alpha \in \Gamma \\
& p^{*a_1} = \sum_{a_1 \in A_1} \lambda^{a_1, s'_1} \quad & a_1 \in A_1 \text{ and } s'_1 \in S_1 \\
& \sum_{a_1 \in A_1} p^{*a_1} = 1.
\end{align*}
\] (2)

Compared with the LP in Yan et al. (2023) for solving $[TV^T_{lb}](s_1, b_1)$, the LP (2) includes the additional constraints $v_{s^i_E} \geq \alpha(s_1, s^i_E)$ for $1 \leq i \leq N_b$ (Horák et al., 2023, Section 9.2) to refine the minimax stage strategy in $[TV^T_{lb}](s_1, b_1)$, such that the lower bound from $\alpha_1$ can be kept as the game evolves, since multiple minimax stage strategies for $Ag_1$ may exist and some of them may deviate from $\alpha_1$.

We illustrate how the strategy $\sigma_{init}^{lb}$ is obtained in Fig. 1 (left), where the red and orange circles indicate the current state, with the size of the interior solid circle representing the probability.
When resolving the game locally \(((s_1, b_1), \alpha_1), A_{\theta_1}\) (blue) plays an action \(a_1\) sampled from a stage strategy \(u_1^{lb}\) computed via (2). Simultaneously, \(A_{\theta_2}\) (red) plays an action \(a_2\) sampled from a stage strategy \(u_2\), which is unknown to \(A_{\theta_1}\). The game moves to the next state and consequently \(A_{\theta_1}\) observes a new agent state \(s_1' \in S_1\). Based on \(a_1, s_1'\) and an assumed stage strategy \(u_2^{lb}\) for \(A_{\theta_2}\), see lines 7–8 of Algorithm 1, \(A_{\theta_1}\) updates its belief to \((s_1', b_1^1, a_1^1, u_1^{lb}, s_1')\) via Bayesian inference, and generates \(\alpha^*_{a_1} \in \text{Conv}(\Gamma)\) from a solution to (2), which forms a new pair for the next resolving.

**Remark 1** There are two key properties of Algorithm 1. First, LP (2) admits at least one solution as \(V_0^{lb}\) is computed by the one-sided NS-HSVI. Second, the current state has to be in the support of the current belief, i.e., \(A_{\theta_1}\) does not lose track of the current state. Such a belief is called a proper belief and this is ensured by assuming the uniform stage strategy for \(A_{\theta_2}\).

**Lemma 1 (Existence and proper belief)** For the NS-HSVI continual resolving at \(((s_1, b_1), \alpha_1)\), the LP (2) admits at least one solution, and if the current state is \((s_1, s_E)\), then \(b_1(s_E) > 0\).

**Proof** The optimal value \(V^*\) has lower and upper bounds \(L = \min_{s \in S, a \in A} r(s, a)/(1 - \beta)\) and \(U = \max_{s \in S, a \in A} r(s, a)/(1 - \beta)\). Let \(V_0^{lb}\) be the lower bound of one-sided NS-HSVI (Yan et al., 2023). Existence follows if (2) admits one solution after any point-based update. The key is to check the feasibility of the first two constraints for \(r_{s_0}^c\) of (2). Initially \(\Gamma' = \{\alpha_0\}\) with \(\alpha_0(s) = L\) for \(s \in S\), and since \(\alpha_1 = \alpha_0\) it is straightforward to verify that (2) admits at least one solution.

For the inductive step, we assume that (2) admits at least one solution for \(\alpha \in \Gamma\) and \(\alpha_1 \in \text{Conv}(\Gamma)\). The point-based update computes a new set \(\Gamma' = \Gamma \cup \{\alpha^*\}\) of PWC functions, where (2) admits at least one solution for \(\alpha_1 = \alpha^*\). Thus, (2) admits at least one solution for \(\alpha \in \Gamma\) and \(\alpha_1 \in \text{Conv}(\Gamma')\). Following the proof of (Horák et al., 2023, Lemma 9.6), we can show that (2) admits at least one solution for \(\alpha \in \Gamma'\) and \(\alpha_1 \in \text{Conv}(\text{Conv}(\Gamma') \cup \{\alpha^*\})\), i.e., \(\alpha_1 \in \text{Conv}(\Gamma')\).

To show the belief is proper, since the initial state is sampled from \((s_1^{init}, b_1^{init})\), which is known to \(A_{\theta_1}\), and the uniform stage strategy for \(A_{\theta_2}\) is assumed, then the result follows.

We next show that our NS-HSVI continual resolving can inherit the soundness of continual resolving, i.e., it can compute an \(\varepsilon\)-minimax strategy for \(A_{\theta_1}\).

**Theorem 1** \((\varepsilon\text{-minimax strategy for }A_{\theta_1})\) The strategy \(\sigma_1^{lb}\) in Algorithm 1 is an \(\varepsilon\)-minimax strategy for \(A_{\theta_1}\) at \((s_1^{init}, b_1^{init})\), i.e., \(\mathbb{E}_{\sigma_1^{lb}, \sigma_2}^{s_1^{init}, b_1^{init}}[Y] \geq V^*(s_1^{init}, b_1^{init}) - \varepsilon\) for all \(\sigma_2 \in \Sigma\).

**Proof** We adapt the proof presented for discrete one-sided POSGs in (Horák et al., 2023, Proposition 9.7). Let \(\mathbb{E}_s[Y]\) denote the expected value of \(Y\) when starting from \(s \in S\) under \(\sigma \in \Sigma\).
Consider any \( b = (s_1, b_1) \in S_B \) and \( \alpha_1 \in \text{Conv}(\Gamma) \). We assume that \( Ag_1 \) follows \( \text{Resolve}_1 \) in Algorithm 1 at the first \( t \) stages and then follows the uniform stage strategy. We denote such a strategy by \( \sigma_1^{b,\alpha_1,t} \). Lemma 1 guarantees that \( \text{Resolve}_1 \) can run for \( t \) stages. We next prove that, for any \( s_E \) with \( b_1(s_E) > 0 \), the expected value of \( Y \) from \( (s_1, s_E) \) under \( \sigma_1^{b,\alpha_1,t} \) has the bound by \( \alpha_1 \):

\[
\mathbb{E}_{\sigma_1^{b,\alpha_1,t}}^{\sigma_2}[Y] \geq \alpha_1(s_1, s_E) - \beta^t(U - L) \quad \text{for all } \sigma_2 \in \Sigma_2. \tag{3}
\]

We prove (3) by induction on \( t \in \mathbb{N} \). For \( t=0 \), (3) holds as \( U \) and \( L \) are trivial lower and upper bounds, and \( \alpha_1 \in \text{Conv}(\Gamma) \) and \( \alpha(s) \leq U \) for all \( s \in S \) and \( \alpha \in \Gamma \). We assume (3) holds for the first \( t \) stages and any \( b' = (s_1', b_1') \in S_B \). The strategy \( \sigma_1^{b,\alpha_1,t+1} \) implies that, at \( (s_1, b_1) \), \( Ag_1 \) takes \( u_{lb} \) (line 4) and then follows the strategy \( \sigma_1^{b',\alpha',t} \) if \( a_1 \) is taken and \( s_1' \) is observed, where \( b' = (s_1', b_1') \) and \( \alpha' = \alpha^{a_1,s_1'} \). Letting \( u_2 \in \mathbb{P}(A_2 | S) \) be the stage strategy of \( Ag_2 \) at \( b \) given by \( \sigma_2 \), using \( b_1(s_E) > 0 \) by Lemma 1, the left side of (3) by replacing \( t \) with \( t + 1 \) equals:

\[
\mathbb{E}_{u_{lb},u_2}[r((s_1, s_E), a)] + \beta\sum_{(a_1, a_2) \in A_1 \land (s_1', s_2') \in S} u_{lb}(a_1)u_2(a_2 | s_1, s_E) \
\cdot \delta((s_1, s_E), (a_1, a_2))(s_1', s_2')\mathbb{E}_{\sigma_1^{b',\alpha',t}}^{\sigma_2}[Y] \quad \text{by definition of } u_{lb} \text{ and } u_2 \text{ and } \delta \\
\geq \min_{a_2 \in A_2} \left( \sum_{a_1 \in A_1} u_{lb}(a_1)r((s_1, s_E), (a_1, a_2)) + \beta\sum_{a_1 \in A_1 \land (s_1', s_2') \in S} u_{lb}(a_1) \right) \\
\cdot \delta((s_1, s_E), (a_1, a_2))(s_1', s_2')\mathbb{E}_{\sigma_1^{b',\alpha',t}}^{\sigma_2}[Y] \quad \text{by linearity in } u_2 \\
\geq \min_{a_2 \in A_2} \left( \sum_{a_1 \in A_1} u_{lb}(a_1)r((s_1, s_E), (a_1, a_2)) + \beta\sum_{a_1 \in A_1 \land (s_1', s_2') \in S} u_{lb}(a_1) \right) \\
\cdot \delta((s_1, s_E), (a_1, a_2))(s_1', s_2')\alpha^{a_1,s_1'}(s_1', s_2') - \beta^t(U - L)) \quad \text{by induction} \\
= \min_{a_2 \in A_2} \left( \sum_{a_1 \in A_1} u_{lb}(a_1)r((s_1, s_E), (a_1, a_2)) + \beta\sum_{a_1 \in A_1 \land (s_1', s_2') \in S} u_{lb}(a_1) \right) \\
\cdot \delta((s_1, s_E), (a_1, a_2))(s_1', s_2')\alpha^{a_1,s_1'}(s_1', s_2') - \beta^{t+1}(U - L) \quad \text{rearranging} \\
\geq \nu_{s_E}^* - \beta^{t+1}(U - L) \quad \text{since } (\nu^*, \lambda_1^*, \mu_1^*) \text{ is a solution to (2) (first constraint)} \\
\geq \alpha_1(s_1, s_E) - \beta^{t+1}(U - L) \quad \text{since } (\nu^*, \lambda_1^*, \mu_1^*) \text{ is a solution to (2) (second constraint)}
\]

and hence (3) holds. Letting \( \sigma_1^{lb} = \lim_{t \to \infty} \sigma_1^{b,\alpha_1,t} \) by definition:

\[
\mathbb{E}_{\sigma_1^{lb}}^{\sigma_2}[Y] = \int_{s_E \in S_E} b_1^{init}(s_E)\mathbb{E}_{\sigma_1^{lb},\sigma_2}^{\sigma_2}[Y]ds_E \\
\geq \langle a^{init}, (s_1^{init}, b_1^{init}) \rangle \quad \text{by (3) and definition of } \langle \cdot, \cdot \rangle \\
= V_{lb}^{\Gamma}(b^{init}) \quad \text{by line 1 of Algorithm 1} \\
\geq V^*(b^{init}) - \varepsilon \quad \text{since } V_{lb}^{\Gamma} \text{ is returned by one-sided NS-HSVI}
\]

which completes the proof.

**4. Inferred-Belief Strategy Synthesis**

We complement our variant of continual resolving with strategy synthesis for \( Ag_2 \), which utilises the upper bound function \( V_{ub}^{\Gamma} \) pre-computed offline and keeps track of an inferred belief about what \( Ag_1 \) believes, which could differ from \( Ag_1 \)'s true belief. Any offline method for fully-observable stochastic games could instead be used, with the associated high computational and storage cost of
generating a complete strategy. Instead, we present an efficient online algorithm, where only one LP is solved at each stage, to synthesise an ε-minimax strategy for Ag2. Since we have pre-computed the offline upper bound, this strategy can guarantee the minimax value from the initial belief, which is known to both agents, but cannot optimally employ the suboptimal actions of Ag1 during play.

**Inferred-belief strategy synthesis.** Our inferred-belief strategy synthesis for Ag2 (Algorithm 2) returns a strategy σ2 ub based on V ub T . The main idea of σ2 ub is that Ag2 keeps a belief (s1, b1), about Ag1’s belief at the current stage, and then computes an action via Resolve2 based on (s1, b1) and V ub T . This belief (s1, b1) is inferred, as Ag2 has no access to what Ag1 actually believes about the state, except the initial belief which is common knowledge. However, since Ag2 is fully-informed, it can simulate an inferred belief update of Ag1, i.e., its belief about Ag1’s next belief.

We illustrate the obtained strategy σ2 ub in Fig. 1 (right). If (s1, b1) is what Ag2 believes about Ag1’s belief and (s, s E) is the current state observed by Ag2, then Ag2 chooses a2 sampled from the stage strategy u2 ub conditioned on (s1, s E) in [TV ub T ](s1, b1) computed via an LP (Yan et al., 2023). At the same time, Ag1 takes a1 ∈ A1 sampled from a stage strategy u1, where Ag2 does not know u1. Then, the game moves to the next state and thus Ag1 observes s′ 1 ∈ S1. Based on a1, s′ 1 and u2 ub, Ag2 updates its belief about Ag1’s belief via Bayesian inference to (s′ 1, b1,a1,u2 ub,s′ 1).

We next show that the inferred-belief strategy is sound, i.e., the inferred-belief carries enough information to generate an ε-minimax strategy for Ag2.

**Lemma 2 (Monotonicity)** If V ub T is an upper bound generated during the one-sided NS-HSVI, then [TV ub T ](s1, b1) ≤ V ub T (s1, b1) for all (s1, b1) ∈ S B.

**Proof** For a given set Υ of belief-value pairs, we first show [TV ub T ] is convex and continuous. From (Yan et al., 2023, Theorem 6 and 7) we have that [TV ub T ](s1, b1) = sup α∈Υ (α, (s1, b1)), where Υ ⊆ F (S) and L ≤ α (s) ≤ U for all s ∈ S and α ∈ Υ. By (Horák et al., 2023, Proposition 4.9), it follows that [TV ub T ](s1, •) is convex. For b1, b′ 1 ∈ P(S E) and α ∈ Υ, using (Yan et al., 2023, Theorem 1), we have ||(α, (s1, b1)) − (α, (s1, b′ 1))|| ≤ K ub(b1, b′ 1), where K ub measures belief difference, and hence ||[TV ub T ](s1, b1) − [TV ub T ](s1, b′ 1)|| ≤ K ub(b1, b′ 1).

Let V ub T and I T be the upper bound and the associated index set after the t-th point-based update, respectively. Similarly to (Horák et al., 2023, Lemma 9.11), we can now prove the monotonicity of [TV ub T ] by induction on t ∈ N. For t = 0, since Υ0 = {((s′ 1, b′ 1), U) ∈ S B × R | i ∈ I 0} for some initial index set I 0. For any (s1, b1) ∈ S B, using (1), we have [TV ub T ](s1, b1) ≤ max s∈S,u∈E α T (s, a) + β U = (1 − β) U + β U = U = V ub T (s1, b1).

For the inductive step, we assume that [TV ub T ](s1, b′ 1) ≤ V ub T (s1, b′ 1) for all (s1, b′ 1) ∈ S B. Thus y i ≥ V ub T (s′ i, b′ i) ≥ [TV ub T ](s′ i, b′ i) for i ∈ I T . Let (s1, b1) ∈ S B be the belief for the
Consider proof and hence by induction $\Upsilon_t$ at stages and then follows the uniform strategy, and denote this strategy by $\Upsilon_t$. By lines 8 and 9 of (Yan et al., 2023, Algorithm 1), we have $V_{ub}^{\Upsilon_t}(s'_t, b'_t) \geq V_{ub}^{\Upsilon_{t+1}}(s'_t, b'_t)$ for all $(s'_t, b'_t) \in S_B$, from which $[TV_{ub}^{\Upsilon_t}](s'_t, b'_t) \geq [TV_{ub}^{\Upsilon_{t+1}}](s'_t, b'_t)$ for any $i \in T^{t+1}$. Therefore we have that $y^* \geq [TV_{ub}^{\Upsilon_t}](s'_t, b'_t) \geq [TV_{ub}^{\Upsilon_{t+1}}](s'_t, b'_t)$ for any $i \in T^{t+1}$.

Now, for any $(s'_t, b'_t) \in S_B$, if $(\lambda'_t)_{i \in T^{t+1}}$ is a solution for $V_{ub}^{\Upsilon_{t+1}}(s'_t, b'_t)$, then by construction:

$$V_{ub}^{\Upsilon_{t+1}}(s'_t, b'_t) = \sum_{i \in T^{t+1}} \lambda'_t \cdot y_i + K_{ub}(b'_t, \sum_{i \in T^{t+1}} \lambda'_t b'_t)$$

$$\geq \sum_{i \in T^{t+1}} \lambda'_t \cdot [TV_{ub}^{\Upsilon_{t+1}}(s'_t, b'_t)] + K_{ub}(b'_t, \sum_{i \in T^{t+1}} \lambda'_t b'_t)$$

$$\geq [TV_{ub}^{\Upsilon_{t+1}}](s'_t, b'_t)$$

since $[TV_{ub}^{\Upsilon_{t+1}}]$ is convex and hence by induction $[TV_{ub}^{\Upsilon}]$ is monotone as required. 

\begin{theorem}[$\varepsilon$-minimax strategy for Ag$_2$] The strategy $\sigma^{ub}_2$ in Algorithm 2 is an $\varepsilon$-minimax strategy for Ag$_2$ at $(s_1^{init}, b_1^{init})$, i.e., $E_{(s_1^{init}, b_1^{init})}^{\sigma_1, \sigma_2} [Y] \leq V^*(s_1^{init}, b_1^{init}) + \varepsilon$ for all $\sigma_1 \in \Sigma_1$.
\end{theorem}

**Proof** Consider $b=(s_1, b_1) \in S_B$. We assume that Ag$_2$ follows Resolve$_2$ in Algorithm 2 for the first $t$ stages and then follows the uniform strategy, and denote this strategy by $\sigma^{b,t}_{2}$. We prove by induction on $t \in \mathbb{N}$ that the expected value of $Y$ from $b$ under $\sigma^{b,t}_{2}$ has the following upper bound:

$$E_{b}^{\sigma^{b,t}_{2}} [Y] \leq V_{ub}^{\Upsilon_t}(b) + \beta^t(U - L)$$

for all $\sigma_1 \in \Sigma_1$. (4)

For $t = 0$, the strategy $\sigma^{b,0}_{2}$ implies that Ag$_2$ adopts the uniform strategy, and therefore (4) directly follows as $U$ and $L$ are lower and upper bounds.

For the inductive step, we assume (4) holds for the first $t$ stages. The strategy $\sigma^{b,t+1}_{2}$ implies that at $b = (s_1, b_1)$, Ag$_2$ takes $u^{ub}_2$ (line 4) and then if $a_1$ is taken and $s'_1$ is observed, follows the strategy $\sigma^{b',t}_{2}$, where $b' = (s'_1, b'_1)$ and $b'_1 = b_{s_1,a_1,u^{lb}_{1}}$. Letting $u_1 \in \mathbb{P}(A_1)$ be Ag$_1$'s stage strategy at $b$ given by any $\sigma_1$, the left-hand side of (4) by replacing $t$ with $t + 1$ equals:

$$E_{b,u_1,u^{lb}_{2}} [r(s, a)] + \beta \sum_{(a_1, s'_1) \in A_1 \times S_1} P(a_1, s'_1 | b, u_1, u^{ub}_2) E_{b'}^{\sigma_1, \sigma^{b',t}_{2}} [Y]$$

$$\leq E_{b,u_1,u^{lb}_{2}} [r(s, a)] + \beta \sum_{(a_1, s'_1) \in A_1 \times S_1} P(a_1, s'_1 | b, u_1, u^{ub}_2) (V_{ub}^{\Upsilon_t}(b') + \beta^t(U - L))$$

by induction

$$\leq E_{b,u^{lb}_{1},u^{ub}_{2}} [r(s, a)] + \beta \sum_{(a_1, s'_1) \in A_1 \times S_1} P(a_1, s'_1 | b, u^{lb}_{1}, u^{ub}_{1}) V_{ub}^{\Upsilon_t}(b') + \beta^{t+1}(U - L)$$

rearranging and since $u^{lb}_{1}$ is a minimax strategy

$$= [TV_{ub}^{\Upsilon_t}](b) + \beta^{t+1}(U - L)$$

by definition of $[TV_{ub}^{\Upsilon_t}]$

$$\leq V_{ub}^{\Upsilon_t}(b) + \beta^{t+1}(U - L)$$

by Lemma 2

as required. Now, letting $\sigma^{ub}_{2} = \lim_{t \to \infty} \sigma^{b,init,t}_{2}$, from (4) we have:

$$E_{b}^{\sigma_1, \sigma^{ub}_{2}} [Y] \leq V_{ub}^{\Upsilon_t}(b^{init}) \leq V^*(b^{init}) + \varepsilon$$

where the last inequality follows from the fact that $V_{ub}^{\Upsilon_t}$ is returned by one-sided NS-HSVI. 

\begin{corollary}[$\varepsilon$-minimax strategy profile] The profile $(\sigma^{lb}_{1}, \sigma^{ub}_{2})$ is an $\varepsilon$-minimax strategy profile.
\end{corollary}
5. Experiments

We evaluated our method on a variant of a pursuit-evasion game (Horák et al., 2023), inspired by mobile robotics (Chung et al., 2011; Isler and Karnad, 2008); see the appendix of Yan et al. (2023) for more detail. The game involves a pursuer, whose aim is to capture an evader. The pursuer is equipped with a ReLU NN classifier, which takes the location (coordinates) of the pursuer as input and outputs one of the 9 abstract grid cells, each consisting of multiple polytopes, with the initial decomposition obtained by computing the preimage of the NN (Matoba and Fleuret, 2020). The pursuer therefore observes which cell it is in, but not its exact location, and knows neither the exact location nor cell of the evader. The evader is fully informed and knows the exact locations of both agents. The evader is captured when both agents are in the same cell. We set a discount of 0.7, reward 100 for capture and timeout of 2h. The (offline) lower and upper bounds for the value of the initial belief are 5.0699 and 6.0665, respectively. We synthesise strategies, which demonstrate that the pursuer can eventually capture the evader with positive probability. Fig. 2 shows stage strategies and lower bounds of states in the belief of the pursuer (top), and evader’s strategies and inferred beliefs (bottom), at different stages. Lower bounds are coloured green and inferred beliefs yellow. The agent positions are highlighted (pink dot for pursuer and light green for evader). The belief of the pursuer and inferred-belief of the evader do not always coincide, e.g., in the third column, the state with bound 2.14 is in the pursuer’s belief, but not the inferred belief. We observe that the pursuer’s strategy selects the moves according to the magnitude of the bound, e.g., in the fourth column, the pursuer moves up or right, since the top right evader position has the highest bound.

6. Conclusions

We have developed an efficient online method to synthesise strategies for a variant of one-sided continuous-state POSGs with discrete observations and validated it on a pursuit-evasion game, in which the partially-informed agent uses a neural network for perception. We have shown that combining continual resolving, inferred beliefs and HSVI bounds computed offline can generate an ε-minimax strategy profile online. For future work, we will consider aggressive assumed stage strategies for the fully-informed agent, since uniform strategies may lead to a large number of states in the belief and consequently large LPs to solve.
Acknowledgements. This project was funded by the ERC under the European Union’s Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No.834115).

References


