Belief Stochastic Game: A Model for Imperfect-Information Games with Known Positions

Achille Morenville and Éric Piette

ICTEAM, UCLouvain, Belgium
{achille.morenville, eric.piette}@uclouvain.be

Abstract. Imperfect-information games present significant challenges for General Game Playing (GGP) agents. Traditional models have limitations that hinder their applicability in this domain. These models require agents to construct and maintain estimates about the game state, a process that is often game-specific and can unintentionally introduce domain-specific knowledge. Furthermore, this specificity undermines the core goal of GGP to generate domain-independent strategies.
To overcome these challenges, we propose the Belief Stochastic Game model. This novel framework shifts the responsibility of state estimation from the agent to the game model itself, allowing agents to focus solely on strategy development. This externalisation of the state estimation process is enabled by the exploitation of the common structures found in many imperfect-information games. The new model facilitates the development of more general agents that can adapt to a wide range of games.

Keywords: General Game Playing · Imperfect-Information Games · Knowledge Representation.

1 Introduction

Imperfect-information games pose a major challenge in Artificial Intelligence and Game Theory. Unlike perfect-information games, where all players know the full game state, these games involve hidden information that players must infer to make optimal decisions. The complexity of this hidden information complicates the development of general strategies and agents.

Traditional models, such as Extensive Form Games (EFG) [16], and more recent ones like Factored-Observation Stochastic Games (FOSG) [11], have been instrumental in representing and analysing imperfect-information games. However, their limitations hinder their use in General Game Playing (GGP). EFG, for example, doesn't differentiate between public and private information, and both EFG and FOSG rely on handcrafted, game-specific state estimation methods. This has led to agents tailored to individual games, making it difficult to generalise strategies across different games.

In this paper, we introduce the Belief Stochastic Game (Belief-SG) model, a novel framework that addresses the limitations of existing models. Belief-SG externalises state estimation, enabling agents to focus entirely on strategy development. By leveraging common structures found in many imperfect-information games, Belief-SG offers a more generalised and standardised approach to reasoning about these games. As a result, it allows the development of agents that can more easily adapt to a wide range of games. This has the potential for many interdisciplinary applications. For example, it could enable historians to simulate and study a wide variety of historical card games, providing new insights into game strategies and decisions [3].

The remainder of this paper is structured as follows: Section 2 provides background and related work, Section 3 introduces the Belief-SG model, Section 4 presents an example and prototype implementation, and we conclude by discussing the implications and future extensions.

2 Background & Related Work

The problems under consideration in this study are imperfect information games. Formally, many models exist to represent a game G. The oldest and most used model is Extensive Form Games (EFG). EFG formalises game states as histories, where a history h is the sequence of actions taken by the players since the beginning of a game. Since some information is hidden from the players, some histories are indistinguishable to them. For example, in poker, players don't know the cards of their opponents, so the histories where the opponents have different cards are indistinguishable. EFG groups these indistinguishable histories into information sets, which are partitions of the possible histories $\mathcal{H}$.

However, this method of representing hidden information overlooks important concepts that are valuable for agents. To play optimally, an agent must consider the available actions of its opponents. In imperfect information games, these actions depend on the knowledge of the opponents, which is not captured by the information sets. Additionally, EFG does not distinguish between publicly available information and information that is private to each player. This distinction, along with the understanding of who knows what, is essential for effective decision making and search strategies in imperfect information games. Furthermore, while there are some works that extend EFG to extract these concepts [4], they rely on hand-crafted solutions that are specific to the games. Moreover, it has been shown that these concepts cannot be extracted in general [12], making EFG unsuitable for general imperfect information games.

In order to address the limitations of EFG, Kovařík et al. proposed a model called Factored-Observation Stochastic Games (FOSG) [11]. This model, a generalisation of the Partially Observable Stochastic Games (POSG) [7], is based on the concept of observation. In FOSG, the agents don't interact directly with the underlying state of the environment. Instead, they receive observations describing the perceivable part of the state. In addition, FOSG divides the observations into private and public parts, allowing agents to reason about what others know. This model can be formalised as a tuple $G = (\mathcal{N}, \mathcal{S}, s_0, \boldsymbol{\mathcal{A}}, T, \boldsymbol{\mathcal{O}}, O, \boldsymbol{R})$ where:

- $\mathcal{N} = \{1, \ldots, N\}$ is a set of N players.

- $\mathcal{S}$ is a set of states and $s_0 \in \mathcal{S}$ is the initial state.
- $\boldsymbol{\mathcal{A}} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N$ is the joint action space, where $\mathcal{A}_i$ being the set of actions of player i.
  - $\mathcal{A}_i(s) \subseteq \mathcal{A}_i$ is the set of legal actions for player i in state s.
  - A state s where $\mathcal{A}_i(s)$ is empty for all players $i \in \mathcal{N}$ is a terminal state.
- $T : \mathcal{S} \times \boldsymbol{\mathcal{A}} \rightarrow \Delta(\mathcal{S})$ is the transition function. After taking a joint action $\boldsymbol{a}$ in state s, the game transitions to a new state $s' \sim T(s, \boldsymbol{a})$.
- $\boldsymbol{\mathcal{O}} = (\mathcal{O}_{priv(1)} \times \ldots \times \mathcal{O}_{priv(N)} \times \mathcal{O}_{pub})$ is the joint observation set. Each player i has a private observation set $\mathcal{O}_{priv(i)}$ and there is a public observation set $\mathcal{O}_{pub}$.
- $O : \mathcal{S} \times \boldsymbol{\mathcal{A}} \rightarrow \Delta(\boldsymbol{\mathcal{O}})$ is the observation function. After taking a joint action $\boldsymbol{a}$ and transitioning to a state s', the observations are sampled according to the observation function $\boldsymbol{o} \sim O(s', \boldsymbol{a})$.
- $\boldsymbol{R} = (R_1, \ldots, R_N)$ is the reward functions where $R_i : \mathcal{S} \times \boldsymbol{\mathcal{A}} \rightarrow \mathbb{R}$ is the reward of player i in a state after a joint action is applied to it.

The game starts in the state s_0. At each turn, each player simultaneously selects a legal action $a_i \in \mathcal{A}_i(s)$ to apply to the current state. The game then transitions to a new state s', sampled according to the transition function $T(s, \boldsymbol{a})$, where $\boldsymbol{a} = (a_1, \ldots, a_N)$ is the joint action. Each player receives a private observation $o_{priv(i)} \in \mathcal{O}_{priv(i)}$ and all players receive a public observation $o_{pub} \in \mathcal{O}_{pub}$ according to the observation function $O(s', \boldsymbol{a})$. The agents also receive a reward based on their reward functions $R_i(s, \boldsymbol{a})$. The game continues until a terminal state is reached. While this formulation is suited to simultaneous games, it can be adapted to sequential games by adding a noop action to the legal actions of players who are not active in the current turn.

Although FOSG provides all the information needed for sound search in imperfect information games, it has limitations that hinder its use in GGP. In FOSG, agents receive only partial observations about the game state, requiring them to construct and maintain state estimates. This often results in handcrafted state estimation tailored to specific games, making it difficult to generalise developments across different games.

The limitations of EFG and FOSG result in the development of agents that are specific to a single game or a small subset of games. For example, Libratus [2], an agent developed for Heads-Up No-Limit Texas Hold'em Poker (HUNL), uses hard-coded abstraction levels designed specifically for HUNL to reduce the state space. This specificity makes it difficult to adapt Libratus's methods to other games. Similarly, DeepStack [14], another HUNL agent, has a state representation tailored to games similar to poker.

While most agents are designed for specific games, some research has taken a GGP approach. One example is Student of Games [20], which has demonstrated strong performance in perfect information games and has outperformed state-of-the-art agents in several imperfect information games. However, this agent is hard-coded to select different architectures based on the game, which restricts its general applicability.

Several systems have been developed to describe and model a wide variety of games. The Stanford General Description Language (S-GDL) [6] is foundational for GGP research, allowing formal game descriptions. Its extension, GDL-II [22], supports imperfect information games but is cumbersome and leaves state estimation to the agent. Ludii [18] simplifies game descriptions, resulting in a large library of games. Although it has been proven that the Ludii language can describe any game [21], its focus on perfect information games limits its application to imperfect information games. Our previous work suggests an extension to Ludii that includes hidden information, aligns more closely with reinforcement learning formalism, and optimises search algorithms to leverage the game description. This approach could enhance Ludii's applicability to general game play in imperfect information settings, broadening its utility beyond perfect information games [15]. While ReCYCLE, with the CardStock simulation engine [1], does support imperfect information games, its design is tailored to classic card games, making it less adaptable for board games like Stratego.

The systems mentioned above are all based on game description languages, but other frameworks use general programming languages to describe games. The most notable is OpenSpiel [13], designed as a flexible framework for reinforcement learning in games. It has a large library of games and is widely used by researchers. However, OpenSpiel is not intended for GGP, as many specific methods must be developed for each game, which introduces domain knowledge. In addition, OpenSpiel still leaves the state estimation to the agent.

3 Model

Although EFG and FOSG offer a strong foundation for reasoning about imperfect information games, their dependence on internal state estimation and representation constrains their applicability in GGP. To overcome these limitations, we introduce a new model, the Belief Stochastic Game, which leverages the structure of games commonly played by humans. By removing the burden of state estimation from the agents, Belief-SG aims to provide a more generalised and standardised framework for reasoning about imperfect information games.

3.1 State Estimation and Representation

The state estimation process in EFG and FOSG is tasked with constructing and updating an estimate of the game state. In imperfect information games, some states are indistinguishable to the agents. To represent this indistinguishability, the estimate, known as the belief state, can be formalised as a probability distribution over all possible game states. A belief state b assigns a probability $b(s)$ to each possible state s, reflecting the likelihood that the game is currently in state s according to the agent's belief.

While it is impractical to find a universal representation of belief states that applies to all games, many games played by humans share common structures that can simplify this challenge. These games often involve different types of pieces such as stones, cards, or other components, which are placed or moved within a defined playable area, such as a board, players' hands, or other regions.

In common imperfect information games, hidden information typically originates from two primary sources:

- the unknown position of the pieces within the playable area (e.g., Battleship, the Kriegspiel variant of Chess, ...).
- the unknown value or identity of the pieces (e.g., the color and suit of a card in Poker, or the rank of a piece in Stratego, ...).

In this paper we focus on the largest category of commonly played games with hidden information, where only the type or identity of the pieces is unknown. Most games involving cards fall into this category, as do board games where all positions are visible, such as Stratego.

Similar to how the state of a game can be described, the belief state must include several key elements: the set of pieces; the playable area, which includes all places where pieces can be placed, such as the board, players' hands, the deck, and other relevant zones; the position of the pieces within this area; the current agents; and any game variables that can change over time, such as an agent's money or the pot in poker. Although it may seem unusual to include the playable area in the state, certain rules or actions in a game can change this area.

In the class of imperfect information games considered here, hidden information arises solely from uncertainty regarding the values of the pieces. Consequently, belief states must also encode probability distributions over the possible values for each piece.

We describe a belief state as a tuple $b = (\mathcal{P}, \mathcal{T}, \mathcal{V}, \theta, \omega, \phi, \Gamma)$ where:

- $\mathcal{P}$ is the set of pieces.
- $\mathcal{T}$ is the set of types that pieces can be.
- $\mathcal{V}$ is the set of all possible values for all types.
  - $\mathcal{V} = \bigcup_{t \in T} \mathcal{V}_t$, where $\mathcal{V}_t \subseteq \mathcal{V}$ is the set of possible values associated with a type $t \in \mathcal{T}$.
- $\theta \colon \mathcal{P} \to \mathcal{T}$ is a function that associates each piece with its type.
- $\omega \colon \mathcal{P} \to 2^{\mathcal{N}}$ is a function that associates each piece with a subset of agents who own it.
- $\phi \colon \mathcal{P} \times \mathcal{V} \to [0, 1]$ is the probability function that associates each piece with a probability distribution over all possible values:
  - Since each piece can only take on values consistent with its type $\phi(p, v) = 0$ for $v \notin \mathcal{V}_{\theta(p)}$.
  - For a valid probability distribution, we require that $\sum_{v \in \mathcal{V}_{\theta(p)}} \phi(p, v) = 1$ for each piece $p \in \mathcal{P}$.
- Γ is a tuple containing additional elements essential for representing the game state, specifically:
  - The playable area of the game, modeled as a graph.
  - A position function, which maps each piece in $\mathcal{P}$ to a specific location within the playable area.
  - The current set of acting players.
  - A mapping of variable names to their current values, representing any additional game-specific parameter.

During the game, the action, or their outcomes may reveal information about the hidden values of the pieces, requiring the belief state to be updated to reflect this information. The update process follows the steps outlined in Algorithm 1.

Algorithm 1: Belief State Update

```
Input: A belief state b, a piece p* ∈ 𝒫, a value v* ∈ 𝒱_θ(p*), and a target
       probability φ* ∈ [0,1]
Result: Set φ(p*, v*) to φ* and propagate this update to other probabilities.
δ* ← φ* − φ(p*, v*)
r* ← 1 − φ(p*, v*)
updated_values = ∅
forall v ∈ 𝒱_θ(p*) \ {v*} do
    u ← δ* · φ(p*, v)/r*
    if u > 0 then
        updated_values ← updated_values ∪ {v}
        φ(p*, v) ← φ(p*, v) − u
φ(p*, v*) ← φ*
r_g ← 0
forall p ∈ 𝒫 \ {p*} do
    if ω(p*) = ω(p) and θ(p*) = θ(p) then
        r_g ← r_g + φ(p, v*)
forall p ∈ 𝒫 \ {p*} do
    if ω(p) ≠ ω(p*) or θ(p*) ≠ θ(p) then
        continue
    δ_p ← − δ* · φ(p, v*)/r_g
    r_p ← Σ_{v ∈ updated_values\{v*}} φ(p, v)
    forall v ∈ updated_values \ {v*} do
        φ(p, v) ← φ(p, v) − δ_p · φ(p, v)/r_p
    φ(p, v*) ← φ(p, v*) + δ_p
```

The algorithm takes as input the belief state b, a target piece $p^* \in \mathcal{P}$, the specific value $v^* \in \mathcal{V}_{\theta(p^*)}$ for which we want to update the probability, and the new probability ϕ^*. The algorithm first adjusts the probability distribution of p^* by setting the probability of v^* to ϕ^* and proportionally updating the probabilities of all other possible values of p^*. Next, the algorithm propagates the change to other pieces that share the same owner and type as p^*. For each of these pieces, the algorithm adjusts the probability distribution over the possible values to remain consistent with the new information about p^*. This propagation step ensures coherence across the belief state, maintaining updated likelihoods for all relevant pieces, taking into account the new information.

Although it may seem that the game master will arbitrarily assign ϕ^*, in practice, updates to piece probabilities occur only when a piece is revealed or when a value becomes impossible. Consequently, the procedure is usually called with ϕ^* set to either 0 or 1, and rarely with other values.

This representation enables encoding belief states across a wide variety of games and different point of view using a unified format, with updates handled by the same procedure. By standardising this process, the state estimation can be shifted from the agent to the game model itself.

3.2 Belief Stochastic Game

Using the previously described belief state representation, we introduce a new model called Belief Stochastic Game (Belief-SG). In this model, the focus shifts from actual states to belief states, as it now handles state estimation. Since each player has different knowledge of the game, the belief states must account for all perspectives. Inspired by the factorisation in FOSG, Belief-SG tracks these viewpoints using three distinct types of belief states:

- The world belief state, b_w, represents the game as if all players had complete information, effectively encoding the actual state of the game. In this belief state, the values of all pieces are fully determined.
- The private belief states, $b_{priv(i)}$ $\forall i \in \mathcal{N}$, represent player i's individual perspective of the game state, based solely on what that player has observed.
- The public belief state, b_{pub}, represents the shared public perspective, containing only information visible to an external spectator.

Formally, we describe a Belief-SG as a tuple $G = (\mathcal{N}, \mathcal{B}, \boldsymbol{B}_0, \boldsymbol{\mathcal{A}}, \tau, \boldsymbol{\rho})$ where:

- $\mathcal{N} = \{1, \ldots, N\}$ is a set of N agents.
- $\mathcal{B}$ is a set of belief states.
- $\boldsymbol{B}_0 = (b_w, b_{priv(1)}, \ldots, b_{priv(N)}, b_{pub}) \subseteq \mathcal{B}$ is a tuple of the initial belief states, where:
  - b_w is the world belief state.
  - $b_{priv(i)}$ represents the private belief state of agent i.
  - b_{pub} represents the public belief state.
- $\boldsymbol{\mathcal{A}} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N$ is the joint action space, where:
  - $\mathcal{A}_i$ being the set of actions of player i.
  - $A_i \colon \mathcal{B} \to \Delta(\mathcal{A}_i)$ is the legal actions function for player i which gives the probability distribution over the actions of player i, encoding its legal actions in a belief state.
- $\tau \colon \mathcal{B} \times \boldsymbol{\mathcal{A}} \to \Delta(\mathcal{B})$ is the transition function that returns a probability distribution over the belief states based on the joint action taken.
- $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)$ is the reward functions, where $\rho_i \colon \mathcal{B} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ is the reward of player i in a belief state after a joint action is applied to it.

Unlike the FOSG and EFG models, the available actions for each player in a belief state are not fully determined. The legal action function must return a probability distribution over possible actions, as the opponents' piece types—and thus their actions—are uncertain. However, as a player knows the values of his own pieces, the legal actions in his private belief state are fully determined and remain identical to those in the actual game state.

The game unfolds in a manner similar to FOSG. A key difference is that the agent only receives its private state $b_{priv(i)}$ along with the public state b_{pub}, rather than observations. Agents also have access to the transition and legal action functions. In addition, when an agent takes an action, all the belief states stored in the game are updated. Since only the global belief state represents the actual state of the game, for each private belief state and the public belief state, the transition that is most consistent with the global belief state is preserved.

The Belief-SG model can be derived from an original FOSG model, similar to the derivation of Belief-MDP from a POMDP [8], ensuring that work designed for FOSG can be applied to Belief-SG. To simplify notations, let $O(\boldsymbol{o} \mid s', \boldsymbol{a})$ denote the probability of observing $\boldsymbol{o}$ given the world state s' and the joint action $\boldsymbol{a}$ in the original FOSG. Similarly, let $T(s' \mid s, \boldsymbol{a})$ represent the probability of transitioning to s' given the world state s and the joint action $\boldsymbol{a}$, and $b(s)$ represent the probability of the world state s in the belief state b. The joint action space $\boldsymbol{\mathcal{A}}$ is the same in both models. The transition probability $\tau(b' \mid b, \boldsymbol{a})$ in the Belief-SG model can be defined as follows:

$$\tau(b' \mid b, \boldsymbol{a}) = \sum_{\boldsymbol{o} \in \boldsymbol{\mathcal{O}}} P(b' \mid b, \boldsymbol{a}, \boldsymbol{o}) P(\boldsymbol{o} \mid b, \boldsymbol{a}),$$

where:

$$\begin{aligned}
P(\boldsymbol{o} \mid b, \boldsymbol{a}) &= \sum_{s' \in \mathcal{S}} O(\boldsymbol{o} \mid s', \boldsymbol{a}) \sum_{s \in S} T(s' \mid s, \boldsymbol{a}) b(s), \\
P(b' \mid b, \boldsymbol{a}, \boldsymbol{o}) &= \begin{cases} 1 & \text{if } \forall s' \in S \colon b'(s') = P(s' \mid b, \boldsymbol{a}, \boldsymbol{o}) \\ 0 & \text{otherwise} \end{cases}, \\
P(s' \mid b, \boldsymbol{a}, \boldsymbol{o}) &= \frac{O(\boldsymbol{o} \mid s', \boldsymbol{a}) \sum_{s \in S} T(s' \mid s, \boldsymbol{a}) b(s)}{P(\boldsymbol{o} \mid b, \boldsymbol{a})}.
\end{aligned}$$

And the reward function $\rho_i(b, \boldsymbol{a})$ in the Belief-SG model can be defined as $\sum_{s \in S} R_i(s, \boldsymbol{a}) b(s)$.

The Belief-SG model offers a more general and unified framework for reasoning about imperfect information games within the GGP domain. By externalising the state estimation process, it enables agents to focus solely on strategy development, addressing key limitations that have previously impeded the advancement of general agents.

4 Example

To illustrate the Belief-SG model, we consider a miniature version of Stratego. We will outline the initial setup and demonstrate how belief states are initialised and updated, showcasing state estimation, transition function, legal actions, and terminal functions.

The game is played by 2 players on a 5×5 board, with each player having 5 pieces: a flag, a bomb, a miner, and two sergeants. The rules are similar to Stratego[1]: capturing the flag wins the game, the bomb kills any attacking piece, the miner defuses the bomb, sergeants capture any piece except the bomb, and only the miner and sergeants can move one square orthogonally. Each player starts with all pieces in their hand.

To initialise the game, the world belief state, private states and public states are created. Each state includes the playing area graph, which in this case is a 5×5 grid of orthogonally connected nodes, plus two additional nodes representing the players' hands, connected to their respective home rows. The pieces are initially placed in the players' hands, and player 0 is set as the starting player. In these belief states, the probability distributions over a player's pieces are uniform if the player is not an observer for that state, while observers fully know their own pieces.

To demonstrate how the game functions, consider the scenario depicted in Figure 1, where each player has already positioned their pieces on the board and made three moves. The red player has already captured a blue piece.

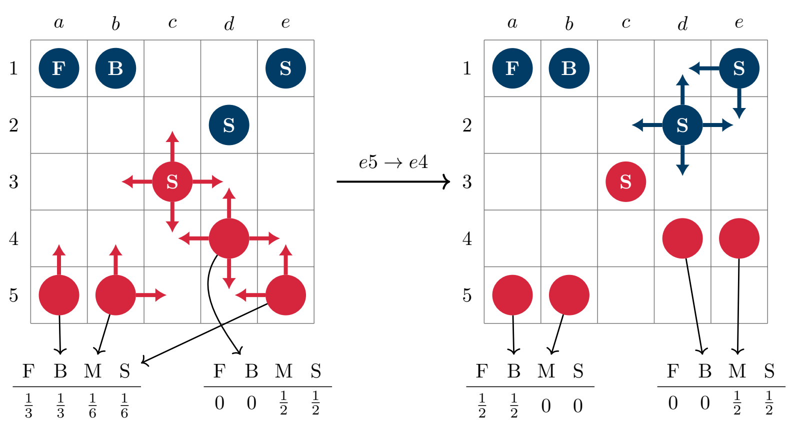

Fig. 1. Belief states in a miniature version of Stratego from the blue player's perspective. Probability distributions over piece types are shown, with known types displayed on the pieces. Legal actions for the respective players are represented by coloured arrows. The transition due to the red player's action $e5 \to e4$ is illustrated.

During the setup phase, legal actions involve placing pieces from the hand onto the player's home row. Once the game begins, legal actions consist of possible moves by miners and sergeants. For pieces that could potentially be a miner or sergeant, legal actions are generated with a probability based on the likelihood of the piece being that type. The set of available actions and their associated probabilities in a specific game state are shown in Figure 1.

[1] https://boardgamegeek.com/boardgame/1917/stratego

The transition function updates belief states based on actions taken. It first applies the action to the pieces, their positions, the graph or the variables, and then updates the probabilities on the values of the pieces. For example, if a piece is moved after the setup phase, the transition function eliminates the possibility that the piece is a flag or a bomb, and also propagates this information to other pieces, as shown in Figure 1.

After each action, the terminal function is called. In this example, a belief state is considered terminal when a player has no piece that could potentially be a flag, or when the acting player has no more valid actions. If the belief state is terminal, the return function awards +1 to the winner and −1 to the other player.

A prototype of the Belief-SG has been implemented in C++ and is freely available[1]. The implementation allows for defining games and agents, is easily extensible, and includes a miniature version of Stratego.

5 Discussion

Externalising the state estimation process from the agent has several important advantages. One major benefit is that it allows the agent to focus entirely on strategy development, without the need to track or estimate hidden information. This streamlines the agent's role, allowing it to focus on maximising rewards based on the knowledge available. In fact, agents are no longer tied to specific game environments or user-defined state estimation, making it easier to apply them to different games and scenarios. This effectively removes the limitations of the EFG and FOSG models and facilitates the development of more versatile agents that can adapt to a wide range of games.

Moreover, by managing imperfect information within the model itself, the agent's perspective is more closely related to that of a model designed for perfect information games. This similarity simplifies the adaptation of methods typically conceived for perfect information games, facilitating their adaptation to apply to games with hidden information.

Another important advantage is that this approach separates the agent's strategic decision-making from its estimation of hidden states. By isolating the policy computation from state estimation, the focus remains on evaluating the quality of policies and strategies. This results in a more objective assessment of agent performance and ensures a fairer comparison between agents.

However, this approach is not without its limitations. The generalisation comes at the cost of increased computational complexity. Since state estimation is handled within the model itself, the complexity of the transition function grows, potentially leading to slower performance when calculating state transitions. This can, in turn, slow down the policy search process, especially in large or complex environments. Additionally, the model is currently suited only for games where hidden information arises from the unknown identities of the pieces. While this

[1] https://github.com/AchilleMorenville/Belief-SG

encompasses many popular games, it does not generalise to all games with hidden information.

6 Conclusion

This paper introduced the Belief Stochastic Game (Belief-SG) model, a framework for reasoning about imperfect information games in the GGP domain. By externalising state estimation, the model allows agents to focus purely on strategy development, addressing limitations that have hindered the development of general agents capable of performing well across diverse games. We also presented a prototype implementation of the Belief-SG model, showcasing its application through a simplified version of Stratego.

Future work will aim to extend the Belief-SG model to address additional sources of hidden information, such as unknown piece positions. We also plan to apply the model to a broader range of games and scenarios, integrating it into existing GGP systems to evaluate its performance and scalability. One promising avenue is to integrate it with Ludii by extending its ludemic game description language with concepts specific to games with imperfect information [19]. In addition, this integration will provide access to a large database of games [5], enabling the interdisciplinary study of historical card games. We will also investigate combining the model with constraint-based methods from perfect-information GGP [9, 10] to evaluate their effectiveness in games with imperfect information.

Acknowledgments. This article is based on the work of COST Action CA22145 - GameTable [17], supported by COST (European Cooperation in Science and Technology).

References

1. Bell, C., Goadrich, M.: Automated playtesting with recycled cardstock. Game & Puzzle Design **2**(1) (2016)
2. Brown, N., Sandholm, T.: Superhuman AI for Heads-up No-limit Poker: Libratus beats top professionals. Science **359**(6374), 418–424 (2017)
3. Browne, C., Soemers, D.J.N.J., Piette, É., Stephenson, M., Conrad, M., Crist, W., Depaulis, T., Duggan, E., Horn, F., Kelk, S., Lucas, S.M., Neto, J.P., Parlett, D., Saffidine, A., Schädler, U., Silva, J.N., de Voogt, A., Winands, M.H.M.: Foundations of Digital Archæoludology. Tech. rep., Schloss Dagstuhl Research Meeting, Germany (2019)
4. Burch, N., Johanson, M., Bowling, M.: Solving imperfect information games using decomposition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 28 (2014)
5. Crist, W., Stephenson, M., Piette, E., Browne, C.: The ludii games database: A resource for computational and cultural research on traditional board games (2024)
6. Genesereth, M.R., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. AI Magazine **26**(2), 62–72 (2005)

7. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: AAAI. vol. 4, pp. 709–715 (2004)
8. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial intelligence **101**(1-2), 99–134 (1998)
9. Koriche, F., Lagrue, S., Piette, É., Tabary, S.: Constraint-based symmetry detection in general game playing. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 280–287 (2017)
10. Koriche, F., Lagrue, S., Piette, É., Tabary, S.: Woodstock : Un programme-joueur générique dirigé par les contraintes stochastiques. Revue D'Intelligence Artificielle (RIA) **31**(3), 307–336 (2017)
11. Kovařík, V., Schmid, M., Burch, N., Bowling, M., Lisỳ, V.: Rethinking formal models of partially observable multiagent decision making. Artificial Intelligence **303**, 103645 (2022)
12. Kovařík, V., Lisỳ, V.: Problems with the efg formalism: a solution attempt using observations. arXiv preprint arXiv:1906.06291 (2019)
13. Lanctot, M., Lockhart, E., Lespiau, J.B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., Hennes, D., Morrill, D., Muller, P., Ewalds, T., Faulkner, R., Kramár, J., Vylder, B.D., Saeta, B., Bradbury, J., Ding, D., Borgeaud, S., Lai, M., Schrittwieser, J., Anthony, T., Hughes, E., Danihelka, I., Ryan-Davis, J.: OpenSpiel: A framework for reinforcement learning in games. CoRR **abs/1908.09453** (2019), http://arxiv.org/abs/1908.09453
14. Moravčík, M., Schmid, M., Burch, N., Lisỳ, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., Bowling, M.: Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science **356**(6337), 508–513 (2017)
15. Morenville, A., Piette, É.: Vers une approche polyvalente pour les jeux à information imparfaite sans connaissance de domaine. In: RJCIA (2024)
16. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton, NJ, USA (1944)
17. Piette, É., Crist, W., Soemers, D.J.N.J., Rougetet, L., Courts, S., Penn, T., Morenville, A.: Gametable cost action: Kickoff report. ICGA Journal **46**(1), 1–17 (2024)
18. Piette, É., Soemers, D., Stephenson, M., Sironi, C.F., Winands, M., Browne, C.: Ludii – the ludemic general game system. In: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020). vol. 325, pp. 411–418 (2020)
19. Piette, É., Stephenson, M., Soemers, D.J.N.J., Browne, C.: General board game concepts. In: Proceedings of the 2021 IEEE Conference on Games (CoG). pp. 932–939. IEEE (2021)
20. Schmid, M., Moravčík, M., et al.: Student of games: A unified learning algorithm for both perfect and imperfect information games. Science Advances **9**(46), eadg3256 (2023)
21. Soemers, D.J., Piette, É., Stephenson, M., Browne, C.: The ludii game description language is universal. In: 2024 IEEE Conference on Games (CoG). pp. 1–8. IEEE (2024)
22. Thielscher, M.: A general game description language for incomplete information games. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 994–999. AAAI (2010)