Mixture of Public and Private Distributions in Imperfect Information Games.

Jérôme Arjonilla[1] **Tristan Cazenave**[1] **Abdallah Saffidine**[2]

[1] LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France
[2] University of New South Wales, Sydney, Australia
jerome.arjonilla@lamsade.dauphine.fr
tristan.cazenave@lamsade.dauphine.fr
abdallah.saffidine@gmail.com

Résumé

Dans les jeux à information imparfaite (par exemple, le bridge, le skat, le poker), l'une des considérations fondamentales est de déduire l'information manquante tout en évitant de divulguer des informations privées. Ne pas tenir compte de l'information privée révélée peut conduire à des performances très exploitables. Toutefois, une attention excessive à cet égard conduit à des hésitations qui ne sont plus cohérentes avec nos informations privées. Dans notre travail, nous montrons que, pour améliorer les performances, il faut choisir d'utiliser ou non l'information privée d'un joueur. Nous étendons notre travail en proposant une nouvelle distribution de croyances en fonction de la quantité d'informations privées et publiques souhaitée. Nous démontrons empiriquement une augmentation des performances et montrons que, afin d'améliorer encore les performances, la nouvelle distribution devrait être utilisée en fonction de la position dans le jeu. Nos expériences ont été réalisées sur plusieurs jeux à information imparfaite et dans plusieurs algorithmes basés sur la détermination (PIMC et IS-MCTS).

Abstract

In imperfect information games (*e.g.* Bridge, Skat, Poker), one of the fundamental considerations is to infer the missing information while at the same time avoiding the disclosure of private information. Disregarding the issue of protecting private information can lead to a highly exploitable performance. Yet, excessive attention to it leads to hesitations that are no longer consistent with our private information. In our work, we show that to improve performance, one must choose whether to use a player's private information. We extend our work by proposing a new belief distribution depending on the amount of private and public information desired. We empirically demonstrate an increase in performance and, with the aim of further improving performance, the new distribution should be used according to the position in the game. Our experiments have been done on multiple benchmarks and in multiple determinization-based algorithms (PIMC and IS-MCTS).

1 Introduction

Search in artificial intelligence has been constantly evolving over the last few decades, and game-oriented research has always been a cornerstone of this success. Chess, Go [14], Poker [2], Skat, Contract Bridge, or Starcraft [21] are among the most famous ones.

Perfect information games (Chess, Go) — where all information is available for each player — have been the most studied, and many algorithms have been able to achieve a level far beyond the level of a human professional player. On the other hand, Imperfect Information Games (IIGs) (Poker, Skat, Bridge) — where some information is hidden — have been less studied, and only a few algorithms are capable of beating professional human player [20, 2, 13].

In IIGs, the complexity is heightened by the missing information, as one must try to infer the missing information of the opponents and, at the same time, be wary to not reveal private hidden information to opponents. Among the methods used in IIGs, determinization-based algorithms — where the hidden information is fixed according to a belief distribution — such as Perfect Information Monte Carlo (PIMC) [11], Recursive PIMC [7], Information Set MCTS [5] or AlphaMu [4] achieve state-of-the-art performance in many trick-taking card games (Contract-Bridge, Skat).

In the work cited above, the determinization operates by sampling the hidden information according to the private information of a given player, *i.e.* what has happened since the beginning, from the point of view of a given agent. However, by doing so, one can indirectly reveal private infor-

mation to opponents, which can lead to a highly exploitable performance.

Recently, the concept of public knowledge [9] — where a distinction is made between observations accessible to everyone and those accessible individually — has emerged in the IIGs. This concept has resulted in many breakthroughs thanks to the decomposition, which made the calculations feasible [12, 1]. Despite this large benefit, there are limitations to its use, especially in the context of belief distribution. By completely removing the knowledge observed by the acting player, one might wonder whether or not using the private information was useful.

In this work, we analyze the impact of using one method rather than another. We also present a new belief distribution, which is a mixture of both public and private belief distribution. We extend the study by analyzing different mixtures, depending on the position within the game. Our experiments are carried out on determinization-based algorithms, which use the belief distribution to fix the incertitude.

The paper is organized as follows : the second section presents notation and current determinization-based algorithms ; Section 3 explains the different belief distributions used with their advantages and drawbacks, and presents our new belief distribution ; Section 4 empirically shows that using the new belief distribution allows us to improve past performance and the last section summarizes our work and future work.

2 Notation and Background

2.1 Notation

We use the notation based on factored-observation stochastic games (FOSGs [9]). This formalism distinguishes between private and public observations.

A game G is composed with a set $\mathcal{N} = \{1, 2, \ldots, N\}$ agents ($N \in \mathbb{N}$). The state of the game is called a **world state** $w \in \mathcal{W}$ and, in each world state, the acting player i chooses an action $a \in \mathcal{A}(w)$, where $\mathcal{A}(w)$ denotes the legal actions at w. After an action a is chosen, we reach the next world state w' from the probability distribution of playing a in w.

During the transition from w to w' by playing a, two observations are received : a **public observation** and a **private observation**. Public observation is the observation visible by every player noted $o_{pub} \in \mathcal{O}_{pub}(w, a, w')$ where $\mathcal{O}_{pub}(w, a, w')$ refers to all the possible public observations. Private observation is the observation visible by a precise player i, noted $o_i \in \mathcal{O}_i(w, a, w')$ where $\mathcal{O}_i(w, a, w')$ refers to all the possible private observations.

A *history* is a finite sequence of legal actions and world states, denoted $h^t = (w^0, a^0, w^1, a^1, ..., w^t)$. For describing the point of view of an agent i of a history h, we introduce an **infostate** $s_i(h)$. An **infostate** for agent i is a sequence of an agent's observations and actions $s_i^t = (\tilde{o}_i^0, a_i^0, \tilde{o}_i^1, a_i^1, ..., \tilde{o}_i^t)$ where $\tilde{o}_i^k = (o_{pub}^k, o_i^k)$. A **public infostate** is a sequence of public observations $s_{pub}^t = (o_{pub}^0, o_{pub}^1, ..., o_{pub}^t)$.

Determinization refers to the fact that we sample one world state according to a belief distribution of the different world states possible. Determinizing the belief distribution is not new and a similar concept exists in other formalisms such as belief state in POMDPs problems [17], occupancy-state in Dec-POMDPs problems [6].

2.2 Determinization-based algorithms

Each determinization-based algorithm has its own characteristics. Nevertheless, they share some common features such as (i) sampling a world state according to a belief distribution over the possible world states, and (ii) using a perfect information algorithm for estimating the value of the sampled world state.

The algorithms are simple and, in practice, they achieve great results, mainly due to the use of perfect information algorithms that are fast and efficient. Among the most famous perfect information algorithms, there are AlphaBeta [8], MCTS [3] or Value Network [14, 15, 16].

In the following, we present two determinization-based algorithms that are baseline and will, at a later stage, be used in our experiments.

2.2.1 PIMC

Perfect Information Monte Carlo (PIMC) is the state of the art of many IIG problems such as Contract-Bridge, Skat, and others.

The algorithm is defined in Algorithm 1 and works as follows (i) samples a world state by using the player's private information ; (ii) plays every action of the sampled world state ; (iii) estimates the new world state by using an algorithm available in perfect information setting ; (iv) repeats until the budget is over ; (v) selects the action that produces the best result in average. In practice, PIMC often uses AlphaBeta as the perfect information evaluator.

2.2.2 IS-MCTS

Information Set Monte Carlo Tree Search (IS-MCTS) [5] uses Monte Carlo Tree Search (MCTS) [3] according to a sampled world state.

MCTS is a state-of-the-art tree search algorithm in perfect information games. It works as follows (i) **selection** — selects a path of nodes based on an exploitation policy ; (ii) **expansion** — expands the tree by adding a new child node ; (iii) **playout** — estimates the child node by using an exploration policy ; (iv) **backpropagation** — backpropagates the result obtained from the playout through the nodes chosen

Algorithm 1: PIMC

```
Function PIMC(s) :
    for m ∈ Moves (s) do
        score[m] ← 0;
    end
    while budget do
        w ← InfoSampling(s);
        for m ∈ Moves (w) do
            score [m] ← score[m] + PerfectAlgo (w,
              m);
        end
    end
    return Best action on average
```

during the selection phase. In practice, MCTS often uses random playout as the perfect information evaluator, and UCB1 in the selection phase.

IS-MCTS works in a similar way to MCTS, but instead of building a tree on world states, IS-MCTS builds a tree on infostate. Yet, at each new iteration, it samples a world state according to using the player's private information and determines the dynamics with this world state (the selection and playout are done on the sampled world state).

Algorithm 2: IS-MCTS

```
Function IS-MCTS(s):
    while budget do
        w ← InfoSampling(s);
        MCTS conditioned on w.;
    end
    return Normalise visit count for each action
Function MCTS(w):
    u ← Selection(w);
    u ← Expansion (u,w);
    u ← Simulation (u,w);
    Backpropagation(u);
```

3 Belief Distributions

To present the different belief distributions, with their advantages and drawbacks, we use the following example throughout the section to facilitate understanding.

The example is based on the famous game 'Liar's Dice' (an explanation of the game is given in Subsection 4.1.2). In our case, two players play, each with 1 die of 2 sides. We denote $\{P_1 : X; P_2 : Y\}$ for player 1 has X and player 2 has Y. There are four world states possible ($w_1 = \{P_1 : 1; P_2 : 1\}$, $w_2 = \{P_1 : 1; P_2 : 2\}$; $w_3 = \{P_1 : 2; P_2 : 2\}$, $w_4 = \{P_1 : 2; P_2 : 1\}$).

For each player, there are two infostates possible and one

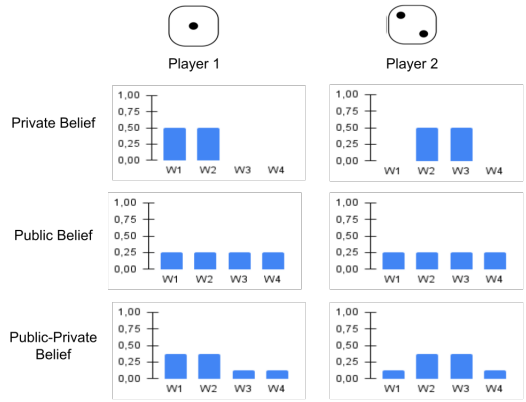


FIGURE 1 – Multiple belief distributions for the game Liar's Dice with 1 2-side die per player. Four world states possible $w_1 = \{P_1 : 1; P_2 : 1\}$, $w_2 = \{P_1 : 1; P_2 : 2\}$, $w_3 = \{P_1 : 2; P_2 : 2\}$ and $w_4 = \{P_1 : 2; P_2 : 1\}$. The Public-Private belief uses the mixture distribution with $\lambda = 0.5$.

public infostate $s_{pub} = \{o^1 = \emptyset, o^2 = \emptyset\}$ (no observation). For the player 1 we have $s_1 = \{\tilde{o}^1 = 1, \tilde{o}^2 = \emptyset\}$ or $s'_1 = \{\tilde{o}^1 = 2, \tilde{o}^2 = \emptyset\}$ (*i.e.* the player 1 observes the die it rolled but not the die rolled by the other player), and for the player 2, we have $s_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1\}$ or $s'_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2\}$ (*i.e.* the player 2 observes the die it receives but not the die received by the other player).

In the following, we suppose that the world state of this example is w_2. Therefore, for the player 1, the infostate is s_1 with two world states possible ($\{w_1, w_2\}$) and for the player 2, the infostate is s'_2 with two world states possible ($\{w_2; w_3\}$).

Figure 1 represents the different belief distributions presented throughout the section.

3.1 Private Distribution

As previously introduced, current determinization-based algorithms work by sampling world states according to the player's private information distribution, *i.e.* knowing a player's private and public observation, we sample a world state.

Let $S_j(s_i)$ be the set of possible infostate for player j conditioning to the infostate s_i of the player i.

In our example, the infostate possible for the player 2 when the player 1 has s_1 is $S_2(s_1) = \{s_2; s'_2\}$. In other world, having the die 1 for the player 1 does not exclude the player 2 to have a 1 or a 2. Yet, depending on the game, this can be restrictive, *e.g.* in trick-taking card games, if the player i has the card 'Queen of Hearts', no opponent can have it.

Definition 1 (Private Belief State) *Let $S_j(s_i)$ be the set of possible infostate for player j conditioning to the infostate s_i. Let $\Delta S_j(s_i)$ denotes the probability distribution*

over the elements of $S_j(s_i)$. We define the private belief state as $\Delta_i(s_i) = (\Delta S_1(s_i), \ldots, \Delta S_i(s_i), \ldots, \Delta S_N(s_i)) = (\Delta S_1(s_i), \ldots, s_i, \ldots, \Delta S_N(s_i))$.

In Figure 1, using Player 1's private belief state provides the following belief distribution $\Delta_1(s_1) = (\{s_1 : 100\%\}, \{s_2 : 50\%; s'_2 : 50\%\})$, which results in two equiprobable world states (w_1, w_2).

When using this distribution for determinization, the algorithm samples a world state (w_1 or w_2) consistent with the current player's information (s_1) and, as the state of the art in trick-taking games shows, great performance is obtained. Yet, by doing so, 3 problems arise.

(i) It is not consistent with the other player's belief, *e.g.* if we use it with the first player, the algorithm samples w_1 or w_2 but never w_3, which is nevertheless, a world state from the point of view of the player 2.

(ii) It is not able to mislead others. In our example, two actions are possible for the first player, 'I have a one' and 'I have a two'. The action 'I have a two' is a lie, however, one may want to play this action with the aim of deceiving the opponent. However, in our case only w_1 or w_2 can be sampled and, in each world, the action 'I have a two' results in a defeat because the second player will say 'This is a lie'. Therefore, lying is never an option, as it never succeeds.

(iii) It, indirectly, allows the opponents to infer our private information, *e.g.* after playing multiple matches, the second player understands that, if the first player plays 'I have a two', it is because he really has a two as it can not lie, and therefore, play to counter it.

Trying to infer the missing information is one of the key components of IIGs, and using the private belief distribution could result in a highly exploitable performance. To remove this problem, one can use public belief distribution, as presented in the next section.

3.2 Public Distribution

Recently in IIGs, many algorithms [12, 1] have been using the concept of public observation. This concept has resulted in many breakthroughs thanks to decomposition, which made the calculations feasible. One application of public observation is the creation of a public belief distribution over the world states possible according to the public observations observed so far.

Definition 2 (Public Belief State [1]) *Let $S_j(s_{pub})$ be the set of possible infostate for player j conditioning to the public infostate s_{pub}. Let $\Delta S_j(s_{pub})$ denote the probability distribution over the elements of $S_j(s_{pub})$. We define the public belief state as* $\Delta_{pub}(s_{pub}) = (\Delta S_1(s_{pub}), ..., \Delta S_N(s_{pub}))$.

In our example, using the public belief state from the point of view of the player 1 or player 2 would result in the same belief distribution $\Delta_{pub} = (\{s_1 : 50\%; s'_1 : 50\%\}, \{s_2 : 50\%; s'_2 : 50\%\}$. Indeed, the public infostate does not contain any information, therefore every world state is possible and equiprobable.

Using a public belief distribution instead of a private belief distribution removes the problem defined in Section 3.1.

(i) It is consistent with the other player's doubts, *e.g.* it samples the world w_3 which is a world state possible of the second player.

(ii) It is capable of misleading others, *e.g.* when sampling w_3 or w_4 the action 'I have a two' does not result in a defeat for the first player, therefore, allows the first player to play the action 'I have a two'.

(iii) It no longer reveals private information, *i.e.* as the reasoning is no longer biased toward the private information, it can not be used against it.

Nevertheless, using public distribution has a significant drawback. It does not consider a player's private information, and one might wonder whether it is useful to not use private information. In Figure 1, when using the public distribution, every world has the same probability, and this, for each player.

It is straightforward to consider that the extent to which private information should be kept hidden depends on the game being played and ; in certain games, it is not necessary to keep the information concealed.

In addition, by using public distribution, one must be cautious that there are more world states possible (*e.g.* by using private distribution, we have two world states possible and by using public distribution, we have four world states possible), which can be intractable in large games.

3.3 Mixture between public and private distribution

To solve both of the problems defined in Section 3.1 and in Section 3.2, we propose to use a mixture of private and public distribution.

Definition 3 (Public-Private Belief State (PPBS)) *Let s_{pub} be the public infostate associated with the infostate s_i. We define the public-private belief state as* $\Delta_\lambda(s_i) = (1 - \lambda)\Delta_i(s_i) + \lambda\Delta_{pub}(s_{pub})$

When $\lambda = 0$, we obtain the private belief distribution, and when $\lambda = 1$, we obtain the public belief distribution.

Using PPBS allows us to be consistent with the problem encountered. When care must be taken not to reveal information, one can increase λ. In contrast, when it is not appropriate to withhold information, one can decrease λ.

In our example, when using PPBS with $\lambda = 0.5$ for the player 1, we obtain the following belief distribution $\Delta_{0.5}(s_1) = (\{s_1 : 75\%; s'_1 : 25\%\}, \{s_2 : 50\%; s'_2 : 50\%\}$,

so that w_1 and w_2 are more probable (37.5% each) than w_3 and w_4 (12.5% each). Nevertheless, their probabilities are not zero, which makes it consistent with the other player's belief.

It is possible to expand this concept by considering that λ depends on the progress of the game. As an example, in trick-taking card games, it may be important to keep the private information hidden at the beginning of the game (so as not to reveal information) but, as the game progresses, the focus shifts to accumulating points before the end, where the importance of concealing this information may decrease.

With the aim of using the public and the mixture distribution, one must take care to adapt the algorithm. In particular, if an algorithm starts at an infostate s_i, it must be adapted to start at the s_{pub}, where s_{pub} is the public infostate associated with the infostate s_i. Adaptations of PIMC and IS-MCTS are given in the appendix.

4 Experimentation

4.1 Benchmarks

For our experiments, the following benchmarks are tested 'Liar's Dice' (LD) and 'Leduc Poker' (LP). Each of them is described below.

4.1.1 Trick-Taking card game

For the purpose of the experimentation, we use a smaller version of classic trick-taking card games. The game is played with two players with N cards divided into four suits (Diamond, Spade, Club, Heart).

The playing phase is decomposed into tricks, the player starting the trick is the one who won the previous trick. The starting player of a trick can play any card in his hand, but the other players must follow the suit of the first player. If they can not, they can play any card they want but, without the possibility of winning the trick. The winner of the trick is the one with the highest ranking card.

At the end of the game, the points of each player are counted. The count is defined by the number of tricks won (plain version of trick-taking card game). A player wins if it has at least half of the points.

4.1.2 Liar's Dice

Liar's dice is a dice game played with two or more players, where each player possesses N K-sided dice, in which a player must deceive and be able to detect an opponent's deception.

In the beginning, each player rolls his dice and observes the values. After that, players take turns guessing the number of dice of a particular type held by everyone. The game continues until a player accuses another of lying. If the player who made the assumption is right, he wins the game, on the opposite, if the challenged player did not lie, the challenged player wins. During the game, a player can not bid less than previously, *i.e.* he must at least bid more dice than the previous player's bid, or the same number of dice, but with a higher value. Lastly, the highest face is a wild card, *i.e.* the value can be used to count for any other face.

4.1.3 Leduc Poker

Leduc Poker, as described in the work [19], is a variation of poker that uses a deck with only two suits, each containing three cards.

The game consists of two rounds. In the first round, each player is dealt a single private card. In the second round, a single board card is revealed. The maximum number of bets allowed is two, with the first round allowing raises of 2 and the second round allowing raises of 4. Both players begin the first round with 1 already in the pot.

4.2 Experimentation

In our experiments, our objective is (i) to observe the extent to which an algorithm X reveals information according to mixture belief distribution ; (ii) to analyze how the mixture belief distribution impacts the performance against an opponent that uses the information revealed ; (iii) to analyze how the mixture belief distribution impacts the performance against an opponent that does not use the revealed information.

Our code is based on OpenSpiel [10]. This is a collection of environments and algorithms for research in general reinforcement learning and search/planning in games.

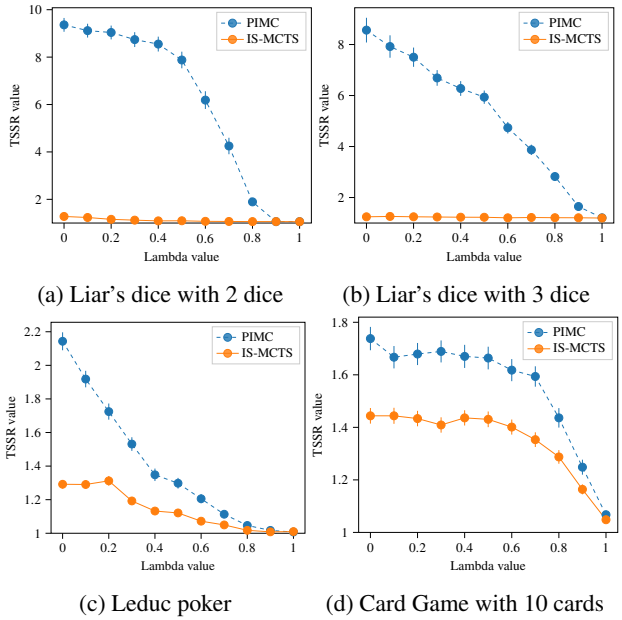
PIMC and IS-MCTS are used with their basic version, *i.e.* PIMC uses AlphaBeta and IS-MCTS uses random rollouts as the perfect information evaluator. For IS-MCTS, the exploration constant is fixed at 0.7. For both, we sample 1000 world states.

To achieve a stable policy (as PIMC and IS-MCTS are online algorithms), we run the algorithm multiple times for every infostate until the policy obtained has less than 1% of variation.

The experiments were conducted according to the player's playing position (each position can reveal more or less information). In the following part, the experiments are carried out for the first player and in the appendix for the second player.

4.2.1 How much information is revealed according λ

We analyze the impact of the information revealed according to the distribution used (public versus private distribution). We use the formula called True State Sampling Ratio (TSSR) [18], which measures how much more likely



(a) Liar's dice with 2 dice (b) Liar's dice with 3 dice

(c) Leduc poker (d) Card Game with 10 cards

FIGURE 2 – Average TSSR according to λ of the mixture distribution when playing at the first position.

it is for the opponent to guess the current world state when using an algorithm X than using a uniform function.

The formula is $TSSR(w) = \eta(w \mid s_i) \cdot |S_i(s_i)|$ where s_i is the infostate corresponding to w, $\eta(w \mid s_i)$ is the probability that the true state is guessed given the information set s_i. The closer the result is to 1, the less likely it is to know the real world state. Figure 2 presents the TSSR value obtained according to λ of the mixture distribution.

As expected, playing closer to the public belief distribution greatly reduces the probability of knowing the real-world state. In 'Liar's Dice' with 2 dice with PIMC, we observe that the opponent is up to 10-fold more likely to guess the real world state when using the private belief distribution instead of the public belief distribution.

'Liar's Dice' reveals more information than 'Leduc Poker'. With 'Liar's Dice', the algorithm reveals up to 10 times more than random, whereas in 'Leduc Poker', it is up to 2 times more than random.

In addition, we observe that PIMC reveals more information than IS-MCTS in every experiment. In 'Liar's Dice' with 2 dice, that is up to 10 times more likely to deduce the true state with PIMC at $\lambda = 0.0$ whereas, with IS-MCTS, it is 'only' 1.5 times more likely to deduce the true state.

4.2.2 How does the mixture impact the performance against the best response

To measure how the mixture impacts the performance, we compute the expected utility against the best responder. The best responder is the worst possible adversary of all algorithms, *i.e.* it knows exactly the policy our algorithm will execute, and therefore, can infer the infostate and plays the best possible action against it.

The results are available in Table 1 where the values represent the expected utility of the best responder and must be minimized. The results obtained are exact utility (without variation), as the best responder computes the best strategy knowing all the distributions in every infostate of the game.

We observe that the private belief distribution tends to perform better than the public belief distribution, *i.e.* for all benchmarks and algorithms (better results are obtained when $\lambda = 0.0$ than when $\lambda = 1.0$).

In 'Liar's Dice' with PIMC, the best performances are obtained when λ is close to 0.5 (with 2 dice, we obtain the best value when $\lambda = 0.6$). These results were expected, as PIMC reveals a lot of information with Liar's Dice, especially when $\lambda < 0.5$, which is then exploited by the best responder.

On the other hand, when the algorithm reveals less information (as observed in 'Leduc Poker' or IS-MCTS), it is preferable to use the private belief distribution or very close, as it is not sufficient for the best responder to exploit the information revealed.

4.2.3 Can the use of multiple mixture belief distributions throughout the game improve performance

We analyze the use of multiple mixtures throughout the game in order to improve the performance. For this purpose, we compute the expected utility against the best responder with multiple λs.

Figure 3 represents several heatmaps for 'Leduc Poker' and 'Liar's Dice' according to the position throughout the game when using PIMC (resp. IS-MCTS). For both games, we have a λ for the first action and another λ for the second action.

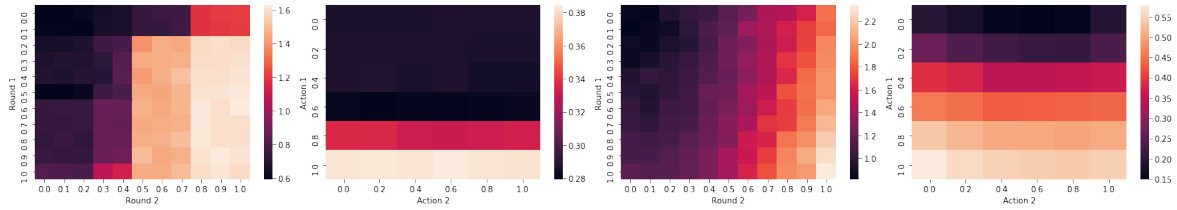
In all figures, we observe that using multiple λ throughout the game has an impact on the performance. In 'Leduc Poker' for both algorithms, not using our private belief distribution is more punished in the second round than in the first round (*e.g.* $(0.0, 1.0)$ has a value of 1.17 whereas $(1.0, 0.0)$ has a value of 1.88 for IS-MCTS).

For 'Liar's Dice', we observe that the first round is the most important one (changing the value of λ in the second round does not have a significant effect on the value obtained).

Furthermore, playing multiple λ can improve performance. In 'Liar's Dice', the best value for IS-MCTS is obtained when we have $\{0.0, 0.6\}$ and for PIMC when we have $\{0.0, 0.6\}$.

Algo	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 2D	0.300	0.298	0.297	0.292	0.294	0.288	**0.281**	0.290	0.336	0.382	0.382
	LD 3D	0.313	0.276	0.265	0.269	**0.235**	0.283	0.324	0.356	0.359	0.393	0.458
	LP	0.622	**0.616**	0.660	0.767	0.797	1.481	1.626	1.480	1.532	1.599	1.611
IS-MCTS	LD 2D	0.513	**0.512**	0.517	0.528	0.539	0.547	0.552	0.554	0.555	0.562	0.562
	LP	**0.797**	0.890	0.966	0.959	1.158	1.226	1.402	1.673	1.786	2.083	2.326

TABLE 1 – Expected utility against best responder when playing at the first player position.



(a) Leduc Poker with PIMC (b) Liar's Dice 2 dice with PIMC (c) Leduc Poker with IS MCTS (d) Liar's Dice 2 dice with IS MCTS

FIGURE 3 – Heatmap of the best response when playing at the first position. Y-axis refers to λ value used when playing the first action, and the X-axis refers to λ used when playing the second action.

4.2.4 How does the mixture impact the winning rate

As observed in the previous experiments, when using a λ closer to the public belief distribution, we obtain a distribution of action less relevant but with the advantage of disclosing less information. Therefore, against an opponent that does not use our private information revealed, it is expected to lose the utility of using λ closer to the public belief distribution.

Nevertheless, using a λ closer to the public belief distribution not only reveals less information but allows it to be more consistent with the hesitation of the other player.

To measure the impact of being more consistent with the other player's hesitation, we evaluate the performance against an algorithm that does not try to infer our private information. To do this, we compute the winning rate against 'PIMC' over 1000 games which results in 3.1% variation (95% of confidence interval). The scores are available in Table 2.

As before, we observe that it is preferable to use the private belief distribution instead of the public belief distribution. For example, in 'Liar's Dice' with 3 dice with PIMC, we observe a drop of 20.8 in the winning rate.

In addition, we observe that in every benchmark tested and for both algorithms, using a λ between 0.0 to 0.5 does not produce a drop in performance, but provides equivalent results. These results are surprising, as we could have expected a drop in performance as the actions are less relevant to the current infostate (as we have sampled less often the true infostate). This implies that being more consistent with the hesitation of the other players compensates for the loss of the player's private information.

5 Conclusion

In this paper, we study the strengths and weaknesses of probability distributions (private and public) in which particular attention has been paid to the information revealed and the impact of this revealed information on performance.

We complete the study by proposing a new probability distribution, a mixture of the two previous ones, which solves problems encountered by other distributions.

We show that using the mixture is beneficial to reduce the information revealed and improve performance. We also show that using multiple mixtures throughout the game improves performance. In addition, we observed that using the mixture against an opponent that does not use our private information revealed can also be used to improve performance, as we are being more consistent with the other player's doubt.

An avenue for improvement would be to extend the utilization of using multiple λ, especially by using a λ at each public infostate. We could also consider using a different lambda for the opponent player.

Another area for improvement would be to extend the study of algorithms that do not use determinization or even, without probability distributions but bearing in mind that one should not always use one's private information at the

Our	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 3D	48.6	**50.4**	47.9	47.4	44.6	42.6	39.9	37.5	36.1	28.9	27.8
	LD 5D	43.1	**43.4**	42.2	**43.4**	42.5	41.4	39.8	36.3	37.1	29.8	23.6
	CG 10C	**48.2**	47.9	47.7	47.7	47.6	47.4	46.7	46.	45.5	39.8	31.6
	CG 20C	53.7	53.8	54.2	**54.5**	53.9	53.2	52.8	52.	47.4	36.3	23.5
IS MCTS	LD 3D	23.7	23.7	24.7	**27.**	23.1	23.1	21.7	20.	19.3	15.4	16.4
	LD 5D	22.	20.9	21.9	**22.2**	21.9	20.8	21.6	21.	16.9	15.5	13.4
	CG 10C	45.3	**46.3**	45.4	45.1	43.8	45.1	45.	43.1	42.7	37.6	30.
	CG 20C	36.5	**38.5**	38.2	36.2	36.4	36.6	35.5	34.9	33.3	33.1	20.8

TABLE 2 – Winning rate when the opponent uses 'PIMC' according to λ of the mixture belief distribution when playing at the first player position.

risk of revealing information and, on the contrary, that one should not always use one's public information in order to be more consistent to one's private knowledge.

Lastly, it would be interesting to extend the results at a larger scale, either by using more games or by using larger games, especially for the calculation of the best responder.

A Appendix

A.1 Adaptation of algorithms

PIMC and IS-MCTS have been created with private belief distribution in mind. Therefore, care must be taken to adjust the algorithms to utilize the public belief distribution or the public-private belief distribution.

To do so, one must use a probability distribution over the infostates possible $S_i(s_{pub})$ before using the algorithm.

A.1.1 PIMC

PIMC requires a distinct algorithm to be applied for each possible infostate, and the final result is obtained by aggregating the scores using the distribution of possible infostates.

A single algorithm is not feasible as the score is calculated based on the actions that are possible for a specific infostate, and not all actions are possible in different infostates.

In the example described in the main article (in Section 3), for the first player, two infostates are possible (s_1 and s'_1). If w_2 is sampled, the algorithm used is the one defined in the infostate corresponding (s_1). In the end, if s_1 has been visited 75% (corresponding to the mixture belief distribution with $\lambda = 0.5$), the action chosen in s_1 will have more impact than the action chosen in s'_1.

A.1.2 IS-MCTS

With IS-MCTS, a singular algorithm is feasible as, IS-MCTS creates a tree where the nodes represent infostates, and an infostate for player j may come from several infostates of player i

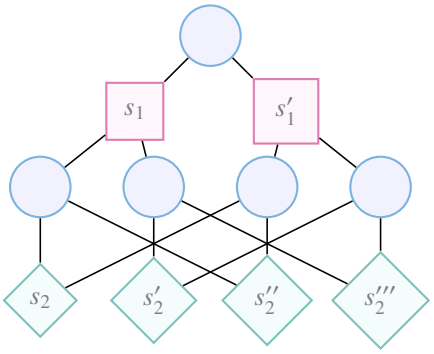


FIGURE 4 – Example of the MCTS tree constructed by IS-MCTS with public belief distribution, when playing the example described in the main article (Section 3). The first player is acting in the red square, the second player is acting in the green diamond and the blue circle refers to the chance node.

An example is provided in Figure 4. In the example, two infostates are possible for the first player (s_1 and s'_1) and four infostates are possible for the second player after the first player's action ($s_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1, \tilde{o}^3 = a_1\}$, $s'_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1, \tilde{o}^3 = a_2\}$, $s''_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2, \tilde{o}^3 = a_1\}$ or $s'''_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2, \tilde{o}^3 = a_2\}$). For the second player, all infostates are achievable through any infostate of the first player, for example, s_2 is achievable when sampling w_1 (from s_1) or when sampling w_2 (from s'_1) and playing the action a_1.

A.2 Complementary experiments

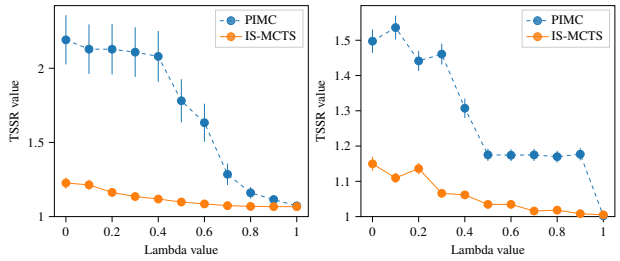
The following experiments are identical to those in the primary paper, with the exception that they are conducted for the second player position.

Algo	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 2D	**0.678**	0.695	0.703	0.707	0.716	0.718	0.711	0.741	0.779	0.836	0.836
	LP	**0.398**	0.400	0.459	0.612	0.796	1.461	1.450	1.509	1.593	1.615	1.632
IS-MCTS	LD 2D	0.697	**0.687**	0.697	0.716	0.727	0.732	0.740	0.751	0.759	0.768	0.787
	LP	**0.784**	0.784	0.898	0.800	1.017	1.078	1.186	1.324	1.561	1.728	2.002

TABLE 3 – Expected utility for best responder against our algorithm being the second player.

Our	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 3D	51.9	**53.**	51.3	49.8	49.7	51.3	48.7	48.6	46.9	46.1	42.7
	LD 5D	**56.7**	55.5	56.	56.2	54.8	56.1	55.3	53.	51.9	44.7	42.3
IS MCTS	LD 3D	48.4	**51.3**	49.9	49.	50.1	51.	47.4	44.	39.7	36.9	33.3
	LD 5D	**48.4**	47.1	48.	46.7	47.8	45.	46.5	40.7	34.4	23.2	14.7

TABLE 4 – Winning rate when the opponent uses 'PIMC' according to λ of the mixture belief distribution when playing at the second player position.



(a) Liar's dice with 2 dice (b) Leduc poker

FIGURE 5 – Average TSSR for IS-MCTS and PIMC on multiple benchmarks according to λ value of the mixture distribution.

Références

[1] Brown, Noam, Anton Bakhtin, Adam Lerer et Qucheng Gong: *Combining Deep Reinforcement Learning and Search for Imperfect-Information Games*. Dans *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc., ISBN 978-1-71382-954-6. event-place : Vancouver, BC, Canada.

[2] Brown, Noam et Tuomas Sandholm: *Superhuman AI for multiplayer poker*. Science, 365 :885 – 890, 2019.

[3] Browne, Cameron, Edward Jack Powley, Daniel Whitehouse, Simon M. M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez Liebana, Spyridon Samothrakis et Simon Colton: *A Survey of Monte Carlo Tree Search Methods*. IEEE Transactions on Computational Intelligence and AI in Games, 4 :1–43, 2012.

[4] Cazenave, Tristan et Véronique Ventos: *The $\alpha\mu$ Search Algorithm for the Game of Bridge*. Dans *Monte Carlo Search at IJCAI*, Communications in Computer and Information Science, 2021.

[5] Cowling, Peter I., Edward Jack Powley et Daniel Whitehouse: *Information Set Monte Carlo Tree Search*. IEEE Transactions on Computational Intelligence and AI in Games, 4 :120–143, 2012.

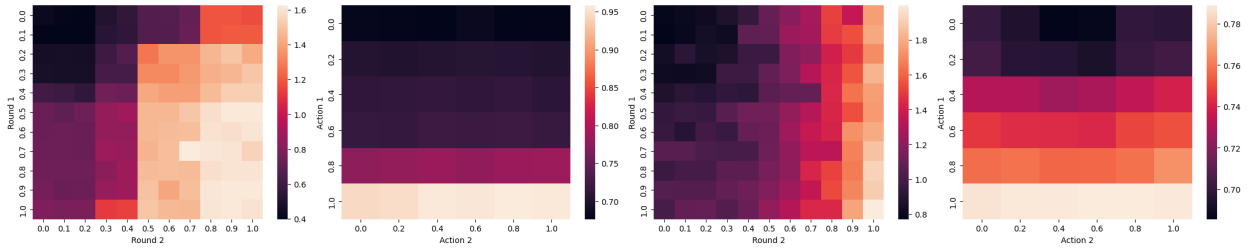
[6] Dibangoye, Jilles Steeve, Christopher Amato, Olivier Buffet et François Charpillet: *Optimally Solving Dec-POMDPs as Continuous-State MDPs*. Journal of Artificial Intelligence Research, 55 :443–497, février 2016.

[7] Furtak, Timothy et Michael Buro: *Recursive Monte Carlo search for imperfect information games*. 2013 IEEE Conference on Computational Inteligence in Games (CIG), pages 1–8, 2013.

[8] Knuth, Donald E. et Ronald W. Moore: *An analysis of alpha-beta pruning*. Artificial Intelligence, 6(4) :293–326, 1975, ISSN 0004-3702.

[9] Kovařík, Vojtěch, Martin Schmid, Neil Burch, Michael H. Bowling et V. Lisý: *Rethinking Formal Models of Partially Observable Multiagent Decision Making*. Artif. Intell., 303 :103645, 2022.

[10] Lanctot, Marc, Edward Lockhart, Jean Baptiste Lespiau, Vinícius Flores Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin



(a) Leduc poker with PIMC (b) Liar's dice 2 dice with PIMC (c) Leduc poker with IS-MCTS (d) Liar's dice 2 dice with IS-MCTS

FIGURE 6 – Heatmap of the best response when playing at the second position. Y-axis refers to λ value used when playing the first action, and the X-axis refers to λ value used when playing the second action.

Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas W. Anthony, Edward Hughes, Ivo Danihelka et Jonah Ryan-Davis: *OpenSpiel : A Framework for Reinforcement Learning in Games*. ArXiv, abs/1908.09453, 2019.

[11] Long, Jeffrey Richard, Nathan R. Sturtevant, Michael Buro et Timothy Furtak: *Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search*. Dans *AAAI*, 2010.

[12] Moravcík, Matej, Martin Schmid, Neil Burch, V. Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, K. Waugh, Michael Bradley Johanson et Michael H. Bowling: *DeepStack : Expert-level artificial intelligence in heads-up no-limit poker*. Science, 356 :508 – 513, 2017.

[13] Perolat, Julien, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis et Karl Tuyls: *Mastering the game of Stratego with model-free multiagent reinforcement learning*. Science, 378(6623) :990–996, dec 2022.

[14] Silver, D., Aja Huang, Chris J. Maddison, A. Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, S. Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel et Demis Hassabis: *Mastering the game of Go with deep neural networks and tree search*. Nature, 529 :484–489, 2016.

[15] Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan et Demis Hassabis: *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. ArXiv, abs/1712.01815, 2017.

[16] Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan et Demis Hassabis: *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*. Science, 362 :1140 – 1144, 2018.

[17] Smith, Trey: *Probabilistic Planning for Robotic Exploration*. Thèse de doctorat, Carnegie Mellon University, Pittsburgh, PA, July 2007.

[18] Solinas, Christopher, Douglas Rebstock et Michael Buro: *Improving Search with Supervised Learning in Trick-Based Card Games*. ArXiv, abs/1903.09604, 2019.

[19] Southey, Finnegan, Michael P Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings et Chris Rayner: *Bayes' bluff : Opponent modelling in poker*. arXiv preprint arXiv :1207.1411, 2012.

[20] Tammelin, Oskari, Neil Burch, Michael Bradley Johanson et Michael Bowling: *Solving Heads-Up Limit Texas Hold'em*. Dans *IJCAI*, 2015.

[21] Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama,

Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps et David Silver: *Grandmaster level in StarCraft II using multi-agent reinforcement learning*. Nature, pages 1–5, 2019.