Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

An Expert-Level Card Playing Agent Based on a Variant of Perfect Information Monte Carlo Sampling

Florian Wisser
Vienna University of Technology
Vienna, Austria
wisser@dbai.tuwien.ac.at

Abstract

Despite some success of Perfect Information Monte Carlo Sampling (PIMC) in imperfect information games in the past, it has been eclipsed by other approaches in recent years. Standard PIMC has well-known shortcomings in the accuracy of its decisions, but has the advantage of being simple, fast, robust and scalable, making it well-suited for imperfect information games with large state-spaces. We propose Presumed Value PIMC resolving the problem of overestimation of opponent's knowledge of hidden information in future game states. The resulting AI agent was tested against human experts in Schnapsen, a Central European 2-player trick-taking card game, and performs above human expert-level.

1 Introduction

Perfect Information Monte Carlo Sampling (PIMC) in tree search of games of imperfect information has been around for many years. The approach is appealing, for a number of reasons: it allows the usage of well-known methods from perfect information games, its complexity is magnitudes lower than the problem of weakly solving a game in the sense of game theory, it can be used in a just-in-time manner even for games with large state-space, and it has proven to produce competitive AI agents in some games. Since we will mainly deal with trick-taking cards games, let us mention Bridge [Ginsberg, 1999], [Ginsberg, 2001], Skat [Buro *et al.*, 2009] and Schnapsen [Wisser, 2010].

In recent years research in AI in games of imperfect information was heavily centered around equilibrium approximation algorithms (EAA). One of the reasons might be the concentration on simple poker variants as in the renowned annual computer poker competition. Wherever they are applicable, agents like Cepheus [Bowling *et al.*, 2015] for heads-up limit hold'em (HULHE) will not be beaten by any other agent, since they are nearly perfect. However, while HULHE has a small state-space ($\sim 10^{14}$), solving this problem still required very substantial amounts of computing power to calculate a strategy stored in 12 terabytes. Our prime example Schnapsen has a state-space around 10^{20}, so even if a near equilibrium strategy was computable within reasonable time, it would take at least 10 exabytes to store it. Using state-space abstraction [Johanson *et al.*, 2013] EAAs may still be able to find good strategies for larger problems, but they depend on finding an appropriate simplification of manageable size. So, we think it is still a worthwhile task to search for in-time heuristics like PIMC that are able to tackle larger problems.

On the other hand, in the 2nd edition (and only there) of their textbook, Russell and Norvig [Russell and Norvig, 2003, p179] quite accurately use the term "averaging over clairvoyancy" for PIMC. A more formal critique of PIMC was given in a series of publications by Frank, Basin, et al. [Frank and Basin, 1998b], [Frank *et al.*, 1998], [Frank and Basin, 1998a], [Frank and Basin, 2001], where the authors show that the heuristic of PIMC suffers from strategy-fusion and non-locality producing erroneous move selection due to an overestimation of MAX's knowledge of hidden information in future game states. A further investigation of PIMC and why it still works well for many games is given by Long et al. [Long *et al.*, 2010], trying to give three easily measurable properties of a game tree, meant to predict the success of PIMC in a game. More recently overestimation of MAX's knowledge is also dealt with in the field of general game play [Schofield *et al.*, 2013]. To the best of our knowledge, all literature on the deficiencies of PIMC concentrates on the overestimation of MAX's knowledge. Frank et al. [Frank and Basin, 1998a] explicitly formalize the "best defense model", which basically assumes a clairvoyant opponent, and state that this would be the typical assumption in game analysis in expert texts. This may be true for some games, but clearly not for all.

Think, for example, of a game of heads-up no-limit hold'em poker playing an opponent with perfect information, knowing your hand as well as all community cards before they even appear on the table. The only reasonable strategy left against such an opponent would be to immediately concede the game, since one will not achieve much more than stealing a few blinds. And in fact expert texts in poker do never assume playing a clairvoyant opponent when analyzing the correctness of the actions of a player.

In the following — and in contrast to the references above — we start off with an investigation of the problem of overestimation of MIN's knowledge, from which PIMC and its known variants suffer. We set this in context to the best defense model and show why the very assumption of it is doomed to produce sub-optimal play in many situations. For

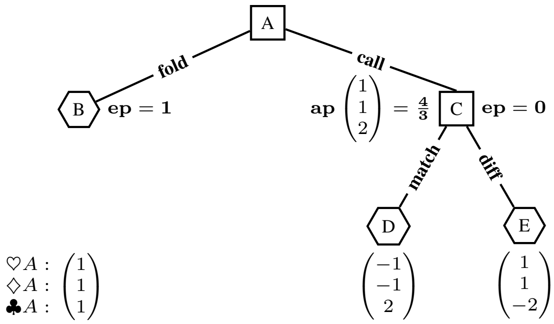

Figure 1: PIMC Tree for XX

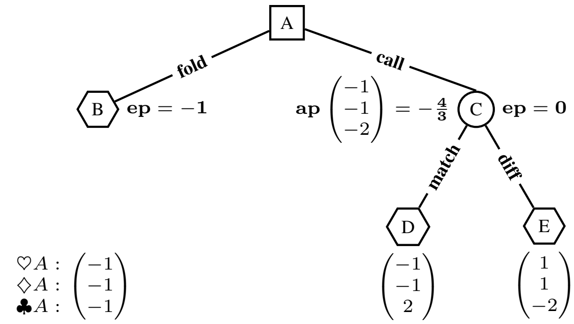

Figure 2: PIMC Tree for XI

the purpose of demonstration we use the simplest possible synthetic games we could think of. We go on defining two new heuristic algorithms, first Presumed Payoff PIMC, targeting imperfect information games decided within a single hand (or leg), followed by Presumed Value PIMC for games of imperfect information played in a sequence of hands (or legs), both dealing with the problem of MIN overestimation. Finally we do an experimental analysis on a simple synthetic game, as well as the Central-European trick-taking card game Schnapsen.

2 Background Considerations

In a 2-player game of imperfect information there are generally 4 types of information: information publicly available (ι^P), information private to MAX (ι^X), information private to MIN (ι^I) and information hidden to both (ι^H). To exemplify the effect of "averaging over clairvoyancy" we introduce two very simple 2-player games: XX with only two successive decisions by MAX, and XI with two decisions, first one by MAX followed by one of MIN. The reader is free to omit the rules we give and view the game trees as abstract ones. Both games are played with a deck of four aces, $\spadesuit A$, $\heartsuit A$, $\diamondsuit A$ and $\clubsuit A$. The deck is shuffled and each player is dealt 1 card, with the remaining 2 cards lying face down on the table. ι^P consists of the actions taken by the players, ι^H are the 2 cards face down on the table, ι^X the card held by MAX and ι^I the card held by MIN.

In XX, MAX has to decide whether to **fold** or **call** first. In case MAX calls, the second decision to make is to bet, whether the card MIN holds matches color with its own card (**match**, both red or both black) or differs in color (**diff**). Fig. 1 shows the game tree of XX with payoffs, assuming without loss of generality that MAX holds $\spadesuit A$. Modeling MAX's decision in two steps is entirely artificial in this example, but it helps to keep it as simple as possible. The reader may insert a single branched MIN node between A and C to get an identically rated, non-degenerate example. Node C is in fact a collapsed information set containing 3 nodes, which is represented by vectors of payoffs in terminal nodes, representing worlds possible from MAX's point of view. To the right of the child nodes B and C of the root node the expected payoff ($\mathbf{ep}$) is given. It is easy to see that the only Nash equilibrium strategy (i.e. the optimal strategy) is to simply fold and cash 1.

To the left of node C, the evaluation of straight PIMC (for better distinction abbreviated by SP in the following) of this node is given, averaging over the payoffs in different worlds after building the point-wise maximum of the payoff vectors in D and E. We see that SP is willing to turn down an ensured payoff of 1 by folding, to go for an expected payoff of 0, by calling and going for either bet then. The reason is the well-known overestimation of hidden knowledge, i.e.: it assumes to know ι^I when deciding whether to bet on matching or differing colors in node C, and thereby evaluates it to an average payoff ($\mathbf{ap}$) of $\frac{4}{3}$. Frank et al. [Frank and Basin, 1998b] analyzed this behavior in detail and termed it strategy-fusion. We will call it MAX-strategy-fusion in the following, since it is strategy-fusion happening in MAX nodes. The basic solution given for this problem is vector minimax. In MAX node C, vector minimax picks the vector with the highest mean, instead of building a vector of point-wise maxima for each world, i.e. the vector attached to C would be either of $(1, 1, -2)$ or $(-1, -1, 2)$, not $(1, 1, 2)$, leading to the correct decision to fold. We list the average payoffs for various agents playing XX on the left-hand side of Table 1. Looking at the results we see that a uniformly random agent (RAND) plays worse than a Nash equilibrium strategy (NASH), and SP plays even worse than RAND. VM stands for vector minimax, but includes all variants proposed by Frank et al., most notably payoff-reduction-minimax, vector-$\alpha\beta$ and payoff-reduction-$\alpha\beta$. Any of these algorithms solves the deficiency of MAX-strategy-fusion in this example and plays optimally.

Let us now turn to the game XI, its game tree given in Fig. 2. The two differences to XX are that MAX's payoff at node B is -1, not 1, and C is a MIN (not a MAX) node. The only Nash equilibrium strategy of the game is MAX calling, followed by an arbitrary action of MIN. This leaves MAX with an expected payoff of 0. Conversely, an SP player evaluates node C to the mean of the point-wise minimum of the payoff vectors in D and E, leading to an evaluation of $-\frac{4}{3}$. So SP always folds, since it assumes perfect knowledge of MIN over ι^X, which is just as wrong as the assumption on the distribution of information in XX. Put in another way, in such a situation SP suffers from MIN-strategy-fusion. VM acts identically to SP in this game and, looking at the right-hand side of Table 1, we see that both score an average of -1, playing worse than NASH and even worse than a random player.

XX	—	XI	ANY
PPP	0	VM	-1
SP	0	SP	-1
RAND	$\frac{1}{2}$	RAND	$-\frac{1}{2}$
NASH	1	NASH	0
VM	1	PPP	0

Table 1: Average Payoffs for XX and XI

World	ι^P	ι^X	ι^I_j	ι^H_j
w_1	—	$\spadesuit A$	$\heartsuit A$	$\diamondsuit A, \clubsuit A$
w_2	—	$\spadesuit A$	$\diamondsuit A$	$\heartsuit A, \clubsuit A$
w_3	—	$\spadesuit A$	$\clubsuit A$	$\heartsuit A, \diamondsuit A$

Table 2: Worlds Evaluated by PIMC in XX and XI

The best defense model [Frank and Basin, 1998a] makes the assumptions that MIN has perfect information (A1), MIN chooses strategy after MAX (A2) and MAX plays a pure strategy (A3). Both, SP and VM (including all subsumed algorithms), implicitly assume at least (A1). And it is this very assumption, that makes them fail to pick the correct strategy for XI, in fact they pick the worst possible strategy. So it is the model itself that is not applicable here. While XI itself is a highly artificial environment, similar situations do occur in basically every reasonable game of imperfect information. Unlike single-suit problems in Bridge, which were used as the real-world case study by Frank et al., even in the full game of Bridge there are situations where the opponent can be forced to make an uninformed guess. This is exactly the situation created in XI, and in doing so, one will get a better average payoff than the best defense assumption suggests.

Going back to the game XI itself, let us for the moment assume that MIN picks its moves uniformly at random (i.e. C is in fact a random node). An algorithm evaluating this situation should join the vectors in D and E with a probability of $\frac{1}{2}$ each, leading to a correct evaluation of the overall situation. And since no knowledge is revealed until MIN has to take its decision, this is a reasonable assumption in this particular case. The idea behind the algorithms proposed in the following is to drop assumption (A1) of facing a perfectly informed MIN, and model MIN instead somewhere between a random agent and a perfectly informed agent.

3 Presumed Payoff PIMC (PPP)

The primary approach of PIMC is to create possible perfect information sub-games in accordance with the information available to MAX. A situation from the point of view of MAX consists of the tuple $S = (\iota^P, \iota^X)$. Fixing some possible states of (ι^I_j, ι^H_j) in accordance with S one gets a perfect information sub-game (or world) $w_j(S) = (\iota^P, \iota^X, \iota^I_j, \iota^H_j)$. Table 2 lists all worlds evaluated in the games XX and XI.

Vector minimax tackles the problem of MAX–strategy–fusion by operating on vectors of evaluations according to the different states of ι^I_j. A similar method to tackle MIN–strategy–fusion in MIN nodes is *not* possible since in every perfect information sub-game the information private to MAX is ι^X, the actual version of information private to MAX usually not known to MIN. So, we have to take a quite different approach.

Recently, Wisser [Wisser, 2013] introduced Error Allowing Minimax (EAM), an extension of the classic minimax algorithm. It defines a custom operator for MIN node evaluation to provide a generic tie-breaker for equally rated actions in games of perfect information. The basic idea of EAM is to give MIN the biggest possible opportunity to make a decisive error. By a decisive error we mean an error leading to a game-theoretically unexpected increase in the games payoff for MAX. To be more specific, the EAM value for MAX in a node H is a triple (m_H, p_H, a_H). m_H is the standard minimax value. p_H is the probability for an error by MIN, if MIN was picking its actions uniformly at random. Finally a_H is the guaranteed advancement in payoff (leading to a payoff of $m_H + a_H$ with $a_H \geq 0$ by definition of EAM) in case of *any* decisive error by MIN. The value p_H is only meant as a generic estimate to compare different branches of the game tree. p_H — seen as an absolute value — does *not* reflect the true probability for an error of a realistic MIN player in a perfect information situation. What it does reflect is the probability for a decisive error by a random player, which serves our purpose perfectly. One of the nice features of EAM is that it calculates its values entirely out of information encoded in the game tree. Therefore, it is applicable to any N-ary tree with MAX and MIN nodes and does *not* need any specific knowledge of the game or problem it is applied to. We will use EAHYB [Wisser, 2013], an EAM variant with very effective pruning capabilities. EAHYB furthermore allows to switch to standard $\alpha\beta$ evaluation at a selectable tree depth given as the number of MIN nodes that should be evaluated with EAM. The respective function $\mathbf{eahyb}$ returns an EAM value given a node of a perfect information game tree.

To define Presumed Payoff Perfect Information Monte Carlo Sampling (PPP) we start off with a 2-player, zero-sum game G of imperfect information between MAX and MIN. As usual we take the position of MAX and want to evaluate the possible actions in an imperfect information situation S observed by MAX. We take the standard approach to PIMC. We create perfect information sub-games

$$w_j(S) = (\iota^P, \iota^X, \iota^I_j, \iota^H_j), \quad j \in \{1, \ldots, n\}$$

in accordance with S. In our implementation w_j are not chosen beforehand, but created on-the-fly using a Sims table based algorithm introduced by Wisser [Wisser, 2010], allowing seamless transition from random sampling to full explorations of all perfect information sub-games, if time to think permits. Let $\mathcal{N}$ be the set of nodes of all perfect information sub-games of G. For all legal actions A_i, $i \in \{1, \ldots, l\}$ of MAX in S let $S(A_i)$ be the situation derived by taking action A_i. For all w_j we get nodes of perfect information sub-games $w_j(S(A_i)) \in \mathcal{N}$ and applying EAHYB we get EAM values

$$e_{ij} := \mathbf{eahyb}(w_j(S(A_i))).$$

The last ingredient we need is a function $k : \mathcal{N} \to [0, 1]$, which is meant to represent an estimate of MIN's knowledge over ι^X, the information private to MAX. For all $w_j(S(A_i))$ we get a value $k_{ij} := k(w_j(S(A_i)))$. While all definitions we make throughout this article would remain well-defined, we still demand $0 \leq k_{ij} \leq 1$, with 0 meaning no knowledge at all and 1 meaning perfect information of MIN. Contrary to the EAM values e_{ij}, which are generically calculated out of the game tree itself, k_{ij} have to be chosen ad hoc in a game-specific manner, estimating the distribution of information. At first glance this seems a difficult task to do, but as we shall see later, a coarse estimate suffices. We allow different k_{ij} for different EAM values, since different actions of MAX may leak different amounts of information. This implicitly leads to a preference for actions leaking less information to MIN. For any pair $e_{ij} = (m_{ij}, p_{ij}, a_{ij})$ and k_{ij} we define the extended EAM value $x_{ij} := (m_{ij}, p_{ij}, a_{ij}, k_{ij})$. After this evaluation step we get a vector $x_i := (x_{i1}, \ldots, x_{in})$ of extended EAM values for each action A_i.

To pick the best action we need a total order on the set of vectors of extended EAM values. So, let x be a vector of n extended EAM values:

$$x = \begin{pmatrix} x_1 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} (m_1, p_1, a_1, k_1) \\ \cdots \\ (m_n, p_n, a_n, k_n) \end{pmatrix} \tag{1}$$

We define three operators $\mathbf{pp}$, $\mathbf{ap}$ and $\mathbf{tp}$ on x as follows:

$$\begin{aligned} \mathrm{pp}(x_j) &:= k_j \cdot m_j + \\ &\quad + (1 - k_j) \cdot \big((1 - p_j) \cdot m_j + p_j \cdot (m_j + a_j)\big) \\ &\quad = m_j + (1 - k_j) \cdot p_j \cdot a_j \\ \mathbf{pp}(x) &:= \frac{\sum_{j=1}^{n} \mathrm{pp}(x_j)}{n} \\ \mathbf{ap}(x) &:= \frac{\sum_{j=1}^{n} m_j}{n} \\ \mathbf{tp}(x) &:= \frac{\sum_{j=1}^{n} \big((1 - p_j) \cdot m_j + p_j \cdot (m_j + a_j)\big)}{n} \end{aligned} \tag{2}$$

The presumed payoff ($\mathbf{pp}$), the average payoff ($\mathbf{ap}$) and the tie-breaking payoff ($\mathbf{tp}$) are real numbers estimating the terminal payoff of MAX after taking the respective action. For any action A with associated vector x of extended EAM values the following propositions hold:

$$\begin{gathered} \mathbf{ap}(x) \leq \mathbf{pp}(x) \leq \mathbf{tp}(x) \\ k_j = 1,\ \forall j \in \{1, \ldots, n\} \Rightarrow \mathbf{ap}(x) = \mathbf{pp}(x) \\ k_j = 0,\ \forall j \in \{1, \ldots, n\} \Rightarrow \mathbf{pp}(x) = \mathbf{tp}(x) \end{gathered} \tag{3}$$

The average payoff is derived from the payoffs as they come from standard minimax, assuming to play a clairvoyant MIN, while the tie-breaking payoff implicitly assumes no knowledge of MIN over MAX's private information. The presumed payoff lies somewhere in between depending on the choice of all $k_j, j \in \{1, \ldots, n\}$.

By the heuristic nature of the algorithm, none of these values is meant to be an exact predictor for the expected payoff ($\mathbf{ep}$) playing a Nash equilibrium strategy, which is reflected by their names. Nonetheless, what one can hope to get is that $\mathbf{pp}$ is a better *relative* predictor than $\mathbf{ap}$ in game trees where MIN-strategy-fusion happens. To be more specific, $\mathbf{pp}$ is a better relative predictor if for two actions A_1 and A_2 with $\mathbf{ep}(A_1) > \mathbf{ep}(A_2)$ and associated vectors x_1 and x_2, $\mathbf{pp}(x_1) > \mathbf{pp}(x_2)$ holds in more situations than $\mathbf{ap}(x_1) > \mathbf{ap}(x_2)$ does.

Finally, for two actions A_1, A_2 with associated vectors x_1, x_2 of extended EAM values we define

$$\begin{aligned} A_1 \leq A_2 :\Leftrightarrow &\big(\mathbf{pp}(x_1) < \mathbf{pp}(x_2)\big) \vee \\ &\big(\mathbf{pp}(x_1) = \mathbf{pp}(x_2) \wedge \mathbf{ap}(x_1) < \mathbf{ap}(x_2)\big) \vee \\ &\big(\mathbf{pp}(x_1) = \mathbf{pp}(x_2) \wedge \mathbf{ap}(x_1) = \mathbf{ap}(x_2) \wedge \\ &\quad \mathbf{tp}(x_1) \leq \mathbf{tp}(x_2)\big) \end{aligned} \tag{4}$$

to get a total order on all actions, the lexicographical order by $\mathbf{pp}$, $\mathbf{ap}$ and $\mathbf{tp}$.

Going back to XI (Fig. 2), the extended EAM-vectors of child nodes B and C of the root node are

$$x_B = \begin{pmatrix} (-1, 0, 0, 0) \\ (-1, 0, 0, 0) \\ (-1, 0, 0, 0) \end{pmatrix}, \quad x_C = \begin{pmatrix} (-1, 0.5, 2, 0) \\ (-1, 0.5, 2, 0) \\ (-2, 0.5, 4, 0) \end{pmatrix}$$

The values in the second and third slot of the EAM entries in x_C are picked by the $\min$ operator of EAM, combining the payoff vectors in D and E. E.g. $(-1, 0.5, 2, 0)$: If MIN picks D MAX looses by -1, but with probability 0.5 it picks E leading to a score of $-1+2$ for MAX. Since the game does not allow MIN to gather any information on the card MAX holds, we set all knowledge values to 0. For both nodes, B and C we calculate their $\mathbf{pp}$ value and get $\mathbf{pp}(x_B) = -1 < 0 = \mathbf{pp}(x_C)$. So contrary to SP and VM, PPP correctly picks to call instead of folding and plays the NASH strategy (see Table 1). Note that in this case $\mathbf{pp}(x_C) = 0$ even reproduces the correct expected payoff, which is not a coincidence, since all parameter in the heuristic are exact. In more complex game situations with other knowledge estimates this will generally not hold. But PPP picks the correct strategy in this example as long as $k_{C1} = k_{C2} = k_{C3}$ and $0 \leq k_{C1} < \frac{3}{4}$ holds, since $\mathbf{pp}(x_B) = -1$ for any choices of k_{Bj} and $\mathbf{pp}(x_C) = -\frac{4}{3}k_{C1}$. As stated before, the estimate on MIN's knowledge can be quite coarse in many situations. To close the discussion of the games XX and XI, we once more look at the table of average payoffs (Table 1). While SP suffers from both, MAX-strategy-fusion and MIN-strategy-fusion, VM resolves MAX-strategy-fusion, while PPP resolves the errors from MIN-strategy-fusion.

We close this section with a few remarks. First, while VM (meaning all subsumed algorithms, including the $\alpha\beta$-variants) increases the computational costs in relation to SP by roughly one magnitude, PPP only does so by a fraction. Second, we checked all operators needed for VM as well as for EAM and it is perfectly possible to redefine these operators in a way that both methods can be blended in one algorithm. Third, while reasoning over the history of a game may expose definite or probabilistic knowledge of parts of ι^I, we still assume all worlds w_j to be equally likely, i.e. we do not model the opponent. If one decides to use such modeling by associating a probability to each world, the operators defined in equation (2) (and in the following equation (5)) can be modified easily to reflect these probabilities.

4 Presumed Value PIMC (PVP)

In most games of imperfect information a match is not decided within a single hand (or leg), e.g. Tournament Poker, Rubber Bridge or Schnapsen, to reduce the influence of pure luck. The strategy picked by an expert player in a particular hand may heavily depend on the overall match situation. To give two examples, expert poker players in a no-limit heads-up match pick very different strategies depending on the distribution of chips between the two players. For certain match states in Rubber Bridge expert texts give the advice to make a bid for a game, which cannot possibly be won, just to save the Rubber.

Presumed Value Perfect Information Monte Carlo Sampling (PVP) takes these match states into account, to decide on when to take which risk. So, for a game let $\mathcal{M}$ be the set of possible match states from MAX's point of view, i.e. the match score between different hands. For each match state $M \in \mathcal{M}$ we need a probability $P(M)$ that MAX is going to win the match from that state. These probabilities might either be estimated ad hoc, or gathered from observation of previous matches. For a match state $M \in \mathcal{M}$ and a payoff m in a particular hand, let M_m be the match state reached, if MAX scores m in this hand. Now, for a vector x of extended EAM values as given in definition (1), we define

$$
\begin{aligned}
\mathrm{pv}(x_j) :=& k_j \cdot P(M_{m_j}) + (1-k_j)\cdot \\
&\cdot\big((1-p_j)\cdot P(M_{m_j}) + p_j \cdot P(M_{m_j+a_j})\big) \\
\mathbf{pv}(x) :=& \frac{\sum_{j=1}^{n} \mathrm{pv}(x_j)}{n} \\
\mathbf{av}(x) :=& \frac{\sum_{j=1}^{n} P(M_{m_j})}{n} \\
\mathbf{tv}(x) :=& \frac{\sum_{j=1}^{n}\big((1-p_j)\cdot P(M_{m_j}) + p_j\cdot P(M_{m_j+a_j})\big)}{n}
\end{aligned}
\tag{5}
$$

PVP works exactly like PPP, except for the operators $\mathbf{pp}$, $\mathbf{ap}$ and $\mathbf{tp}$ defined in (2) being replaced by the operators $\mathbf{pv}$, $\mathbf{av}$ and $\mathbf{tv}$ respectively. In the definition of the total order applied to possible actions in definition (4), the operators are replaced likewise.

The presumed value ($\mathbf{pv}$), the average value ($\mathbf{av}$) and the tie-breaking value ($\mathbf{tv}$) are estimates for the probability to win the match from a match state reached by scoring the respective payoff. $\mathbf{pv}$ takes into account, that the match state reached might not only be M_{m_j}, but the better match state $M_{m_j+a_j}$ with a probability p_j given by the EAM values and depending on the estimated knowledge k_j of MIN. Given that the consistency condition for the probabilities $P(M_{m_j+a_j}) > P(M_{m_j})$ holds, which means that a higher payoff in this hand leads to a match state with a higher probability to win from, all statements in equation (3) hold for the respective operators in equation (5) as well. $\mathbf{pv}$, $\mathbf{av}$ and $\mathbf{tv}$ are now in relation to the expected value $\mathbf{ev}$ to win the overall match taking the action in question. What has been stated for the predictive qualities of $\mathbf{pp}$, $\mathbf{ap}$ and $\mathbf{tp}$ with respect to $\mathbf{ep}$ remains true for $\mathbf{pv}$, $\mathbf{av}$ and $\mathbf{tv}$ with respect to $\mathbf{ev}$.

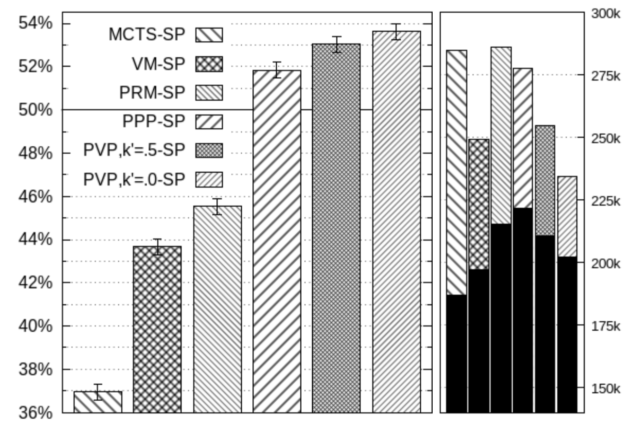

Figure 3: Experimental Results from Kinderschnapsen

5 Trick Taking Card Games — A Case Study

The first case study is performed with a synthetic trick-taking cards game, which we call Kinderschnapsen. It is a stripped down version of Schnapsen. The aim was to simplify the game as far as possible, while leaving enough strategic richness to give an interesting example. Kinderschnapsen is a 2-player, zero-sum game of strategy with imperfect information but perfect recall with an element of chance. Its rules can be found online[1].

We measured the parameters described by Long et al. [Long *et al.*, 2010] to predict the success of SP in this game. The parameters were measured using random play in 100000 games, similar to the methods used for Skat and Hearts by Long et al. We found a bias of $b = 0.33$, a disambiguation factor of $df = 0.41$ and a leaf correlation of $lc = 0.79$, indicating that SP is already strong in this game.

Fig. 3 summarizes the experimental results from Kinderschnapsen. The left-hand side shows different AI agents playing an SP player. Each result was measured in a tournament of 100000 matches. This huge number was chosen in order to get statistically significant results in a game where luck plays an important role. Alongside each result we give the Agresti-Coull confidence interval [Agresti and Coull, 1998] to a confidence level of 98% as error bars, to check statistical significance. The 50% line is printed solid for easier checking of superiority (confidence interval entirely above), inferiority (confidence interval entirely below) or equality (confidence interval overlaps). The knowledge estimate used for PVP was set to $k = k' + (1-k')\frac{h}{N}$, with k' a configurable minimal amount of knowledge of MIN over ι^X, h the number of cards in MAX's hand and N the number of cards MIN has not seen during the game so far. Using $\mathbf{eahyb}$ we set the depth at which it switches to standard $\alpha\beta$ evaluation to 5. The probabilities for winning a match from a given match state needed for PVP is estimated ad hoc, but replaced by the measured frequency during the course of a tournament.

So, in the left-hand side of Fig. 3 we see that PVP with a minimal amount of knowledge set to $k' = 0$ plays best against SP, winning around 53.6% of all matches. This seems

[1] http://www.doktorschnaps.at/index.php?page=ks-rules-english

a humble advantage, but given the simplicity of the game with even a random player winning around 12% against SP and SP already playing the game well, it is a good improvement over SP. PVP with $k' = 0.5$ shows that the knowledge estimate can be chosen quite coarsely. PPP still defeats SP with high statistical significance, but has only a slight advantage. We also adopted and implemented variants of payoff-reduction-minimax (PRM) and vector-minimax (VM), scaling to the game of Kinderschnapsen. While PRM performs slightly better than VM both are inferior to SP. Furthermore we implemented Monte Carlo Tree Search (MCTS), an algorithm *not* following the PIMC approach which has been successfully applied to perfect information as well as imperfect information games (see e.g. [Browne *et al.*, 2012]). After some tests we had to conclude that it does not perform well in Kinderschnapsen against PIMC–based agents. The best MCTS agent, running 1000 iterations at any decision with a UTC parameter set to $\sqrt{2}$ only won 36.9% of its matches against SP.

The right-hand side of Fig. 3 gives one of the reasons, why PVP performs better than PPP, SP, PRM, VM and MCTS. Kinderschnapsen, as well as Schnapsen, allows an action referred to as closing the stock, which allows winning a game by a higher score but failing to win from closing is punished by a high score in favor of the opponent. Looking at the success of a player after closing the stock gives a good insight on the risk a player is willing to take and how good it judges its chances to win from that point on. Each bar gives the number of match points scored after the respective player closed the stock, with the solid black part showing the points won and the shaded part the points lost (mention that the scale of the y-axis starts at 140k). While PPP scores the most points of all agents by closing, it also looses nearly 56k points in doing so. PVP (with $k' = 0$) looses the least points from closing by far with only 32k, still winning over 200k points. Out of all algorithms under consideration, PVP judges its risks best. To close the discussion of Kinderschnapsen let us note that PVP increases evaluation time by roughly 30% compared to SP, while PRM and VM increase it by 600%.

The second test case was the Central-European trick-taking card game Schnapsen, a 2-player, zero-sum game of strategy with imperfect information but perfect recall with an element of chance. Although it is played with only 20 cards, its state-space contains an estimated 10^{20} different states. The size of the state-space moves calculation or approximation of game theoretic optima out of reach. We follow the usual tournament rules[2]. We have chosen this game for two reasons. First, and most importantly, we have access to an online platform (`www.doktorschnaps.at`, DRS) playing Schnapsen against human experts. We were allowed to use our algorithm to back up the AI, to give a reasonably large set of matches against human experts to judge its strength. Second, it is well suited for a PIMC approach as was shown already in a large number of matches against human experts, allowing a comparison of PVP against SP in a field where SP is already a strong challenge for human players.

Fig. 4 shows the results of PVP, PPP and SP against each

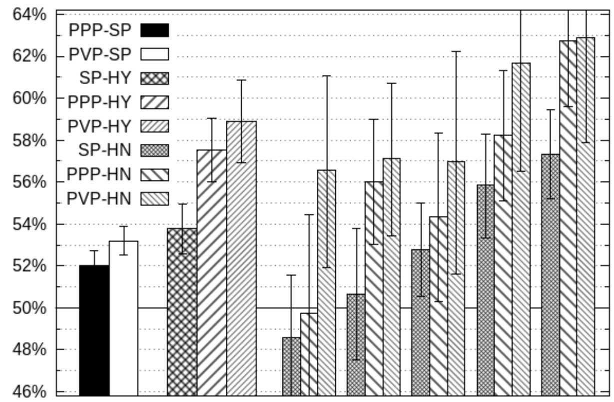

Figure 4: Experimental Results from the Game Schnapsen

other and against human players. For PVP the knowledge was set similarly to Kinderschnapsen with $k' = 0$, an EAM tree depth for $\mathbf{eahyb}$ starting at 1 increasing to ∞ during the course of a hand and probabilities to win a match from a given state compiled from measured frequencies of earlier matches of DRS. The 2 bars on the left show the results of PVP and PPP against the SP agent in tests over 30000 matches (all error bars still to a confidence level of 98%). Both show significant superior performance over SP, PVP showing the larger gain in performance winning slightly above 53.2%.

DRS played against human players by SP until January 2014 ($\sim$ 9500 matches), by PPP from February to September 2014 ($\sim$ 5800) and by PVP from October 2014 to date ($\sim$ 3500). All AI agents are given 6 seconds per decision which may be exceeded once per game if less than 7 perfect information sub-games were evaluated (in most stages of the game all agents respond virtually instantaneously). Human players have a time limit of 3 minutes per action and contrary to usual tournament rules are allowed to look at all cards already seen at any time and their trick points are calculated and displayed in order to prevent loosing by bad memory or miscalculation. All bars besides the 2 rightmost in Fig. 4 are compiled from machine vs. human games with the group of 3 thicker bars representing machine vs. humanity (HY) and each group of 3 thinner bars representing machine vs. a specific human player (HN). We give the results of the best individuals having played enough matches to get results of statistical significance. While SP already showed significant superiority against humanity winning 53.8% of all games, individual human players managed to be on par with it, with the best player loosing only 48.6% against SP. PPP does better than SP against humanity as well as each individual, but is still on par with some players. PVP does best winning 58.9% of its games against humanity and defeating any human player facing it with high statistical significance. This is, PVP did not win less than 56.5% against any human player (including experts in the game) having played enough games to get significant results, which lets us conclude that PVP shows superiority over human expert-level play in Schnapsen. To gain insight on the difference between PVP and SP we looked at over 130k decisions PVP has already taken in matches against

[2]http://www.doktorschnaps.at/index.php?page=rules-english

humans. In 17.4% PVP picks different actions than SP would, with **pv** overruling **av** in 15.9% and **tv** tie-breaking **av** in 1.5% of actions.

A further proof of PVP's abilities was given evaluating 110 interesting situations from Schnapsen analyzed by a human expert[3]. Out of these 110 situations PVP agreed with human expert analysis in 101 cases. In some of the remaining 9 cases PVP's evaluation helped to discover errors in the expert analysis.

6 Conclusion and Future Work

Dropping the assumptions of the best defense model to play versus a clairvoyant opponent increases the performance of SP in trick-taking card games, with PVP playing above human expert-level in Schnapsen. Opposed to equilibrium approximation approaches (EAA) PVP does not need any precalculation phase, nor large amounts of storage for its strategy. It is capable of taking its decision just-in-time in games with large state-spaces.

We implemented PPP for heads-up limit hold'em total bankroll, to get a case study in another field. Unfortunately the next annual computer poker competition (ACPC) will be held in 2016 and we were unable to organize a match up with one of the top agents of ACPC 2014 so far. We implemented a very simple version of a counter factual regret (CFR) agent. CFR was the approach that produced the best agents in the ACPCs of the last years. Our CFR agent uses the coarsest possible game abstraction, so it is only meant as a reference and not as a competitive agent. The average payoff per game of PPP vs. CFR was four times as high as that of SP vs. CFR.

We are looking forward to take part in the annual computer poker competition (ACPC) with PPP/PVP agents to see their performance against top equilibrium approximation agents.

References

[Agresti and Coull, 1998] A. Agresti and B.A. Coull. Approximate is better than exact for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[Bowling *et al.*, 2015] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit holdem poker is solved. *Science*, 347(6218):145–149, 2015.

[Browne *et al.*, 2012] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):1–43, 2012.

[Buro *et al.*, 2009] Michael Buro, Jeffrey Richard Long, Timothy Furtak, and Nathan R. Sturtevant. Improving state evaluation, inference, and search in trick-based card games. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1407–1413, 2009.

[Frank and Basin, 1998a] Ian Frank and David A. Basin. Optimal play against best defence: Complexity and heuristics. In H. Jaap van den Herik and Hiroyuki Iida, editors, *Computers and Games*, volume 1558 of *Lecture Notes in Computer Science*, pages 50–73. Springer, 1998.

[Frank and Basin, 1998b] Ian Frank and David A. Basin. Search in games with incomplete information: A case study using bridge card play. *Artif. Intell.*, 100(1-2):87–123, 1998.

[Frank and Basin, 2001] Ian Frank and David A. Basin. A theoretical and empirical investigation of search in imperfect information games. *Theor. Comput. Sci.*, 252(1-2):217–256, 2001.

[Frank *et al.*, 1998] Ian Frank, David A. Basin, and Hitoshi Matsubara. Finding optimal strategies for imperfect information games. In *AAAI/IAAI*, pages 500–507, 1998.

[Ginsberg, 1999] Matthew L. Ginsberg. Gib: Steps toward an expert-level bridge-playing program. In Thomas Dean, editor, *IJCAI*, pages 584–593. Morgan Kaufmann, 1999.

[Ginsberg, 2001] Matthew L. Ginsberg. GIB: imperfect information in a computationally challenging game. *J. Artif. Intell. Res. (JAIR)*, 14:303–358, 2001.

[Johanson *et al.*, 2013] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 271–278. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[Long *et al.*, 2010] Jeffrey Richard Long, Nathan R Sturtevant, Michael Buro, and Timothy Furtak. Understanding the success of perfect information monte carlo sampling in game tree search. In *AAAI*, 2010.

[Russell and Norvig, 2003] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Second Edition*. Prentice Hall, Upper Saddle River, NJ, 2003.

[Schofield *et al.*, 2013] Michael Schofield, Timothy Cerexhe, and Michael Thielscher. Lifting hyperplay for general game playing to incomplete-information models. In *Proc. GIGA 2013 Workshop*, pages 39–45, 2013.

[Wisser, 2010] Florian Wisser. Creating possible worlds using sims tables for the imperfect information card game schnapsen. In *ICTAI (2)*, pages 7–10. IEEE Computer Society, 2010.

[Wisser, 2013] Florian Wisser. Error allowing minimax: Getting over indifference. In *ICTAI*, pages 79–86. IEEE Computer Society, 2013.

[3] http://psellos.com/schnapsen/blog/