Improving a Case-Based Texas Hold'em Poker Bot

Ian Watson, Song Lee, Jonathan Rubin & Stefan Wender

Abstract - **This paper describes recent research that aims to improve upon our use of case-based reasoning in Texas hold'em poker bot called CASPER. CASPER uses knowledge of previous poker scenarios to inform its betting decisions. CASPER improves upon previous case-based reasoning approaches to poker and is able to play evenly against the University of Alberta's Pokibots and Simbots, from which it initially acquired its case-bases and updates previously published research by showing that CASPER plays profitably against human online competitors for play money. The new research described here shows how CASPER has improved its play against human players by remembering and reusing its own game play history.**

I. INTRODUCTION

The game of poker provides an interesting environment to investigate how to handle uncertain knowledge and issues of chance and deception in hostile environments. Games in general offer a well suited domain for investigation and experimentation due to the fact that a game is usually composed of several well defined rules which players must adhere to. Most games have precise goals and objectives which players must meet to succeed. For a large majority of games the rules imposed are quite simple, yet the game play itself involves a large number of very complex strategies. Success can easily be measured by factors such as the amount of games won, the ability to beat certain opponents or, as in the game of poker, the amount of money won.

Up until recently AI research has mainly focused on games such as chess, checkers and backgammon. These are examples of games which contain perfect information. The entire state of the game is accessible by both players at any point in the game, e.g. both players can look down upon the board and see all the information they need to make their playing decisions. These types of games have achieved their success through the use of fast hardware processing speeds, selective search and effective evaluation functions [1].

Games such as poker on the other hand are classified as stochastic, imperfect information games. The game involves elements of chance (the actual cards which are dealt) and hidden information in the form of other player's hole cards (cards which only they can see). This ensures that players now need to make decisions with uncertain information present.

The focus of this paper is to investigate the application of CBR to the game of poker. We have developed a poker playing softbot, called CASPER (CASe-based Poker playER), that uses knowledge of past poker experiences to make betting decisions.

Dept of Computer Science, University of Auckland, Auckland, New Zealand; email: ian@cs.auckland.ac.nz

CASPER plays the variation of the game known as "limit Texas Hold'em" and has been tested against other poker bots and real players.

The remainder of this paper is structured as follows, section two will detail related previous research, section three gives a brief introduction to the game of Texas hold'em. Sections four, five and six describe the design and implementation of CASPER. This is followed by the experimental results obtained against poker-bots and real players online for both play and real money. The paper concludes with a discussion of the results and the potential for future work.

II. RELATED WORK

Over the last few years there has been a dramatic increase in the popularity of the game of Texas hold'em. This growing popularity has also sparked an interest in the AI community with increased attempts to construct poker robots (or bots), i.e. computerised poker players who play the game based on various algorithms or heuristics. Recent approaches to poker research can be classified into three broad categories:

1. **Heuristic rule-based systems:** which use various pieces of information, such as the cards a player holds and the amount of money being wagered, to inform a betting strategy.

2. **Simulation/Enumeration-based approaches:** that consist of playing out many scenarios from a certain point in the hand and obtaining the expected value of different decisions.

3. **Game-theoretic solutions:** which attempt to produce optimal strategies by constructing the game tree in which game states are represented as nodes and an agents possible decisions are represented as arcs.

The University of Alberta Poker Research Group are currently leading the way with poker related research, having investigated all of the above approaches. Perhaps the most well known outcome of their efforts are the poker bots nicknamed Loki [2] and [3] Poki.

Loki originally used expert defined rules to inform a betting decision. While expert defined rule-based systems can produce poker programs of reasonable quality [3], various limitations are also present. As with any knowledge-based system a domain expert is required to provide the rules for the system. In a strategically complex game such as Texas hold'em it becomes impossible to write rules for all the scenarios which can occur. Moreover, given the dynamic, nondeterministic structure of the game any rigid rule-based system is unable to exploit weak opposition and

is likely to be exploited by any opposition with a reasonable degree of strength. Finally, any additions to a rule-based system of moderate size become difficult to implement and test [4].

Loki was later rewritten and renamed Poki. A simulation-based betting strategy was developed which consisted of playing out many scenarios from a certain point in the hand and obtaining the expected value (EV) of different decisions. A simulation-based betting strategy is analogous to selective search in perfect information games.

Both rule-based and simulation-based versions of Poki have been tested by playing real opponents on an IRC poker server. Poki played in both low limit and higher limit games. Poki was a consistent winner in the lower limit games and also performed well in the higher limit games where it faced tougher opposition [3] .

More recently the use of game theory has been investigated in the construction of a poker playing bot. The University of Alberta Computer Poker Research Group have attempted to apply game-theoretic analysis to full-scale, two-player poker. The result is a poker bot known as PsOpti that is "able to defeat strong human players and be competitive against world-class opponents" [5].

There have also been numerous other contributions to poker research outside the University of Alberta Poker Research Group. Sklansky and Malmuth, [6] and [7], have detailed various heuristics for different stages of play in the game of Texas hold'em. The purpose of these rules, however, has been to guide human players who are looking to improve their game rather than the construction of a computerised expert system. Korb et al., [8] produced a Bayesian Poker Program (BPP) which makes use of Bayesian networks to play five-card stud poker, whilst Dahl investigated the use of reinforcement learning for neural net-based agents playing a simplified version of Texas hold'em [9].

We have encountered relatively few attempts to apply the principles and techniques of CBR to the game of poker. Sandven and Tessem constructed a case-based learner for Texas hold'em called Casey [10]. Casey began with no poker knowledge and builds up a case-base for all hands that it plays. They report that Casey plays on a par with a simple rule-based system against three opponents, but loses when it faces more opponents. Salim and Rohwer have attempted to apply CBR to the area of opponent modeling, i.e., trying to predict the hand strength of an opponent given how that opponent has been observed playing in the past [11]. While CBR seems inherently suited to this particular type of task they report better performance by simply relying on long-term average.

III. TEXAS HOLD'EM

Texas hold'em is the variation used to determine the annual World Champion at the World Series of Poker. This version of the game is the most strategically complex and provides a better skill-to-luck ratio than other versions of the game [6].

The game of Texas hold'em is played in four stages, these include the preflop, flop, turn and the river. During each round all active players need to make a betting decision. Each betting decision is summarised below:

Fold: A player discards their hand and contributes no money to the pot. Once a player folds they are no longer involved in the current hand, but can still participate in any future hands.

Check/Call: A player contributes the least amount possible to stay in the hand. A check means that the player invests nothing, whereas a call means the player invests the least amount required greater than $0.

Bet/Raise: A player can invest their own money to the pot over and above what is needed to stay in the current round. If the player is able to check, but they decide to add money to the pot this is called a bet. If a player is able to call, but decides to add more money to the pot this is called a raise.

All betting is controlled by two imposed limits known as the *small bet* and the *big bet*. For example, in a $10/$20 game the small bet is $10 and all betting that occurs during the preflop and the flop are in increments of the small bet. During the turn and the river all betting is in increments of the big bet, $20. All results detailed in this paper refer to a $10/$20 limit game.

Each of the four game stages are summarised below:
1. **Preflop**: The game of Texas hold'em begins with each player being dealt two hole cards which only they can see. A round of betting occurs. Once a player has made their decision play continues in a clockwise fashion round the table. As long as there are at least two players left then play continues to the next stage. During any stage of the game if all players, except one, fold their hand then the player who did not fold their hand wins the pot and the hand is over.
2. **Flop**: Once the preflop betting has completed three community cards are dealt. Community cards are shared by all players at the table. Players use their hole cards along with the community cards to make their best hand. Another round of betting occurs.
3. **Turn**: The turn involves the drawing of one more community card. Once again players use any combination of their hole cards and the community cards to make their best hand. Another round of betting occurs and as long as there are at least two players left then play continues to the final stage.
4. **River**: During the river the final community card is dealt proceeded by a final round of betting. If at least two players are still active in the hand a showdown occurs in which all players reveal their hole cards and the player with the highest ranking hand wins the entire pot (in the event that more than one player holds the winning hand then the pot is split evenly between these players).

IV. CASPER SYSTEM OVERVIEW

CASPER uses CBR to make a betting decision. This means that when it is CASPER's turn to act he evaluates the current state of the game and constructs a target case to represent this information. A target case is composed of a number of features. These features record important game information such as CASPER's hand strength, how many opponents are in the pot, how many opponents still need to act and how much money is in the pot. Once a target case has been constructed CASPER then consults his case-base (i.e. his knowledge of past poker experiences) to try and find similar scenarios which may have been encountered. CASPER's case-base is made up of a collection of cases composed of their own feature values and the action which was taken, i.e. fold, check/call or bet/raise. CASPER uses the k-nearest neighbour algorithm to search the case-base and find the most relevant cases, these are then used to decide what action should be taken.

CASPER was implemented using the commercially available product Poker Academy Pro 2.51 and the Meerkat API. The University of Alberta Poker Research Group provides various poker bots with the software including instantiations of Pokibot and the simulation based bot Simbot. These poker bots have been used to generate the case library for CASPER. Approximately 7,000 hands were played between various poker bots and each betting decision witnessed was recorded as a single case (or experience) in CASPER's case-base. Both of Alberta's bots have proven to be profitable against human competition in the past [12], so we therefore assume that the cases obtained are of sufficient quality to enable CASPER to play a competitive game of poker.

V. CASE REPRESENTATION

CASPER searches a separate case-base for each separate stage of a poker hand (i.e. preflop, flop, turn and river). The features that make up a case and describe the state of the game at a particular time are listed and explained in [13]. These are the indexed features or case vocabulary, which means that they are believed to be predictive of a case's outcome and by computing local similarity for each feature they are used to retrieve the most similar cases in the case-base. Eight features are used in all case-bases and four features are only used during the postflop stages. Each case has a single outcome that is the betting decision that was made. Case features include: No. of players at the table, Relative position at table, Players in current hand, Players yet to act, Bets committed, Bets to call, Pot Odds, Hand strength, Positive potential, Negative potential, Small bets in pot, Previous round bets, and Action (i.e., the betting decision).

The 'hand strength' feature differs somewhat for preflop and postflop stages of the game. During the preflop there exists 169 distinct card groups that a player could be dealt. These card groups were ordered from 1 to 169 based on their hand ranking, where 1 indicates pocket Aces (the best

preflop hand) and 169 indicates a 2 and a 7 of different suits (the worst preflop hand). Preflop hand strength was then based on this ordering, whereas postflop hand strength refers to a calculation of immediate hand strength based on the hole cards a player has and the community cards which are present. This value is calculated by enumerating all possible hole cards for a single opponent and recording how many of these hands are better, worse or equal to the current player's hand. For more details on hand strength and potential consult [3] and [12].

VI. CASE RETRIEVAL

Once a target case has been constructed CASPER needs to locate and retrieve the most similar cases it has stored in its case-base. The k-nearest neighbour algorithm is used to compute a similarity value for all cases in the case-base. Each feature has a local similarity metric associated with it that evaluates how similar its value is to a separate case's value, where 1.0 indicates an exact match and 0.0 indicates entirely dissimilar.

Two separate similarity metrics are used depending on the type of feature. The first is the standard Euclidean distance function given by the following equation:

$$ si = 1 - \left(\frac{|x_1 - x_2|}{MAX\_DIFF} \right) \qquad (1) $$

where: x_1 refers to the target value, x_2 refers to the case value and MAX_DIFF is the greatest difference in values, given by the range in [13].

The above Euclidean similarity metric (1) produces smooth, continuous changes in similarity, however, for some features, minor differences in their values produce major changes in actual similarity, e.g. the 'Bets to call' feature. For this reason an exponential decay function, given by equation (2), has also been used for some features:

$$ si = e^{-k \, (|x_1 - x_2|)}, \qquad (2) $$

where: x_1 refers to the target value and x_2 refers to the source value and k is a constant that controls the rate of decay.

Global similarity is computed as a weighted linear combination of local similarity, where higher weights are given to features that refer to a player's hand strength as well as positive and negative potential. The following equation (3) is used to compute the final similarity value for each case in the case-base:

$$ \sum_{i=1}^{n} w_i x_i \bigg/ \sum_{i=1}^{n} w_i \qquad , (3) $$

where: x_i refers to the ith local similarity metric in the range 0.0 to 1.0 and w_i is its associated weight, in the range $0 - 100$.

---

[1] http://www.poker-academy.com/

After computing a similarity value for each case in the case-base a descending quick sort of all cases is performed. The actions of all cases which exceed a similarity threshold of 97% are recorded. Each action is summed and then divided by the total number of similar cases which results in the construction of a probability triple (f, c, r) which gives the probability of folding, checking/calling or betting/raising in the current situation. If no cases exceed the similarity threshold then the top 20 similar cases are used. As an example, assume CASPER looks at his hole cards and sees A♥-A♠. After a search of his preflop case-base the following probability triple is generated: (0.0, 0.1, 0.9). This indicates that given the current situation CASPER should never fold this hand, he should just call the small bet 10% of the time and he should raise 90% of the time. A betting decision is then probabilistically made using the triple which was generated.

VII. RESULTS

A. Against the Poker-Bots

CASPER was evaluated by playing other poker bots provided through the commercial software product Poker Academy Pro 2.5. CASPER was tested at two separate poker tables. The first table consisted of strong, adaptive poker bots that model their opponents and try to exploit weaknesses. As CASPER has no adaptive qualities of his own he was also tested against non-adaptive, but loose/aggressive opponents. A loose opponent is one that plays a lot more hands, whereas aggressive means that they tend to bet and raise more than other players. All games were $10/$20 limit games which consisted of 9 players. All players began with a starting bankroll of $100,000. The adaptive table consisted of different versions of the University of Alberta's poker bots: Pokibot and Simbot. Figure 1 records the amount of small bets won at the adaptive table over a period of approximately 20,000 hands. Two separate versions of CASPER were tested separately against the same competition. CASPER02 improves upon CASPER01 by using a larger case-base, generated from approximately 13,000 poker hands. A poker bot that makes totally random betting decisions was also tested separately against the same opponents as a baseline comparison.
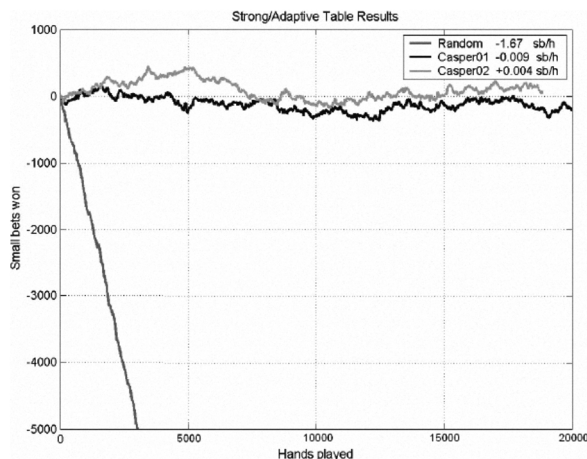

Fig. 1. CASPER vs. strong adaptive poker bots

While CASPER01 concludes with a slight loss and CASPER02 concludes with a slight profit, Figure 1 suggests that both versions approximately break even against strong competition, whereas the random player exhausted its bankroll of $100,000 after approximately 6,000 hands. CASPER01's small bets per hand (sb/h) value is -0.009 which indicates that CASPER01 loses about $0.09 with every hand played. CASPER02 slightly improves upon this by winning approximately $0.04 for every hand played. Random play against the same opponents produces a loss of $16.70 for every hand played.

The second table consisted of different versions of Jagbot, a non-adaptive, loose/aggressive rule-based player. Figure 2 records the amount of small bets won over a period of approximately 20,000 hands. Once again a bot which makes random decisions was also tested separately against the same competition as a baseline comparison for CASPER.
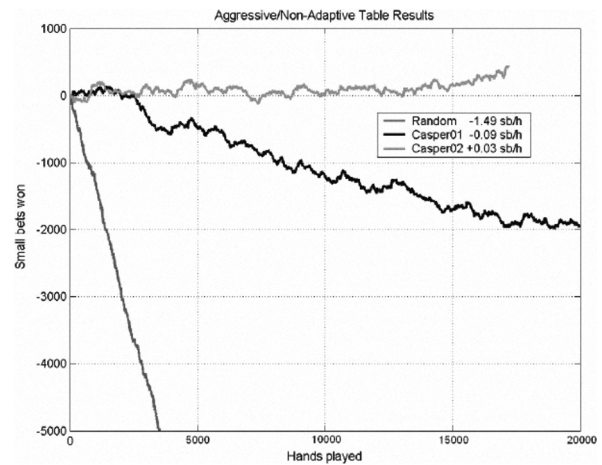

Fig. 2. CASPER vs. aggressive non-adaptive poker bots

Figure 2 indicates that CASPER01 is unprofitable against the non-adaptive players, losing approximately $0.90 for each hand played. CASPER02 shows a considerable improvement in performance. With more cases added to the case-base CASPER02 produces a profit of +0.03 sb/h, or $0.30 for each hand played. Once again the random player exhausted its initial bankroll after approximately 7000 hands, losing on average $14.90 for each hand played.

B Real Opponents - Play Money

CASPER02 was tested against real opponents by playing on the 'play money' tables of internet poker websites. Here players can participate in a game of poker using a bankroll of play money beginning with a starting bankroll of $1000. All games played at the 'play money' table were $10/$20 limit games. At each table a minimum of two players and a maximum of nine players could participate in a game of poker. CASPER was tested by playing anywhere between one opponent all the way up to eight opponents. Figure 3 displays the results recorded at 'play money' tables for CASPER (using hand picked weights) and CASPERGeneral (using feature weights derived by an evolutionary algorithm described in another paper) as well as a random opponent

which makes random decisions (used as a baseline comparison).

Both CASPER and CASPERGeneral earn consistent profit at the "play money" tables. The results suggest that the use of CASPER with hand picked weights outperforms CASPERGeneral. CASPER earns a profit of $2.90 for every hand played, followed by CASPERGeneral with a profit of $2.20 for each hand. Random decisions resulted in exhausting the initial $1000 bankroll in only 30 hands, losing approximately -$30.80 for each hand played.
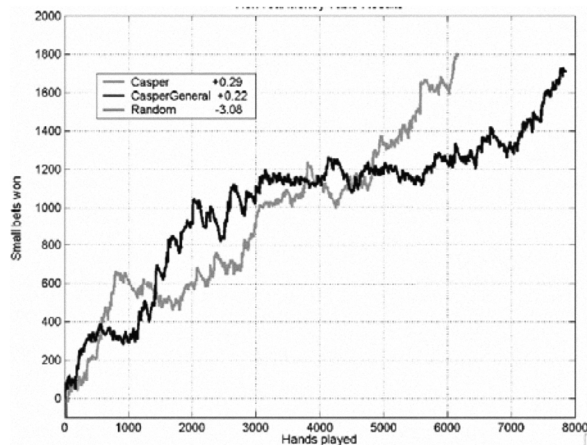

Figure 3. CASPER vs. real opponents for play money

Both Pokibot and Simbot have also been tested against real opponents by playing on Internet Relay Chat (IRC). The IRC server allows bots and humans to challenge each other online using "play money". Results reported by [12] indicate that Pokibot achieves a profit of +0.22 sb/h, i.e. a profit of $2.20 per hand, and Simbot achieves a profit of +0.19 sb/h or $1.90 profit per hand. These results are very similar to those obtained by CASPER, when challenging real opponents for play money. As CASPER used Pokibot and Simbots playing style to build its case-base this result would be expected.

Figure 4. CASPER vs. real opponents for real money

Because CASPER was playing against real opponents the time taken to record the results is longer than when challenging computerised opponents. For this reason, fewer hands were able to be played against real opponents. CASPER is also mainly suited to playing poker at a full table, i.e. with nine or ten players present, however the results recorded above consist of anywhere between two and nine players at a table. While we need to take caution in analysing the above results, it is safe to say that CASPER is consistently profitable at the 'play money' tables.

C. Real Opponents - Real Money

Because there is normally a substantial difference in the type of play at the 'play money' tables compared to the 'real money' tables it was decided to attempt to get an idea of how CASPER would perform using real money against real opponents. CASPER02 (the large case-base) with hand-picked feature weights that had achieved the best performance at the 'play money' tables was used to play at the 'real money' tables. The betting limit used was a small bet of $0.25 and a big bet of $0.50. CASPER started out with a bank roll of $100. The results are given in Figure 4.

CASPER achieves a small bet per hand value of -0.07. Therefore, CASPER now loses on average $0.02 per hand. The results indicate that while CASPER loses money very slowly it is now, nonetheless, unprofitable against these opponents. Due to the fact that real money was being used, fewer hands were able to be played and the experiment was stopped after CASPER had lost approx. $50. No results are available for Pokibot or Simbot challenging real opponents using real money. Therefore, it is not possible to evaluate how CASPER performs using real money compared to Pokibot or Simbot.

VIII. IMPROVING AGAINST REAL PLAYERS

We can assume that CASPER does not play well against real players for real money because the game play of real people for real money is different than that of the Alberta poker-bots from which CASPER obtained its case-based. This assumption is confirmed if we study the average similarity of retrieved cases used to decide betting in each of our experimental scenarios. Similarity in a case-based reasoner can be thought of as being equivalent to accuracy. If cases of high similarity are retrieved their suggested actions (i.e., betting decisions) should be more accurate than if cases of low similarity are retrieved (this is a fundamental assumption of CBR [14]).

The average similarity of cases retrieved when playing against the bots is approx. 90%. The average similarity of cases retrieved against real money players is approx. 50%. This is a clear indication that CASPER is struggling to find useful cases when playing against real people. Indeed the fact that it plays profitably for play money is somewhat surprising given the low similarity and may be a product of the fact that people play recklessly when they are not worried about losing money. They are in effect giving money away by playing with reckless abandon.

CBR theory would inform us that if CASPER uses cases that are more representative of the current playing situation then the retrieval similarity should improve and its performance should be improved [14]. This hypothesis can be tested by having CASPER acquire a case-base from real players. To this end CASPER was modified to retain cases whilst it was playing online. Every time CASPER played an online hand for play money the case was retained. CASPER started out with the original case-base obtained from the Alberta poker-bots and added to it. Figure 4 shows the small bet per hand winnings of CASPER as it acquires 10,000 new cases from real players for play money.
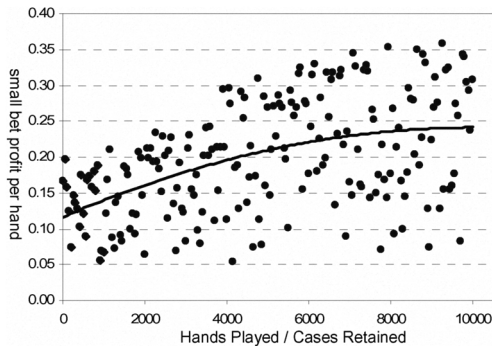

Figure 5. CASPER vs. Real Players for Play Money

As can be seen in Figure 5 CASPER is improving its play from a maximum of $0.20 small bets per hand profit to $0.35 per hand. It also appears from the trend line that around 7000-8000 new cases retained its performance is starting to plateau. The variance is play remains large which is probably a reflection that people play aggressively or recklessly with play money and therefore CASPER is learning to play with less caution.

Figure 6 shows the similarity of cases retrieved where it can be seen that average similarity improves from approx. 50% (0.5) to 70% (0.7) as the case-base grows to 10,000 cases. It is also evident that as the case base grows the similarity of retrieved cases more frequently approaches 1.0. This indicates that the case-base CASPER has acquired is more representative of the cases it is seeing whilst playing for play money against real people.
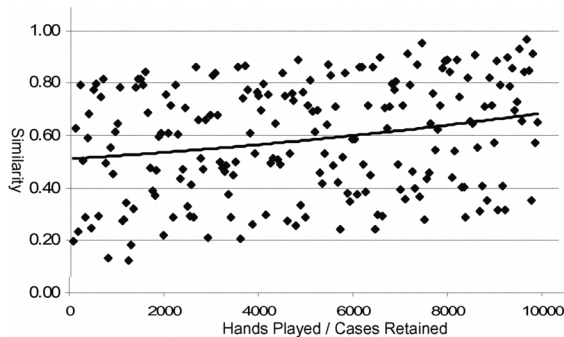

Figure 6. CASPER vs. Real People for Play Money Showing Similarity

IX. CONCLUSION & FUTURE WORK

In conclusion, CASPER, a case-based poker player has been developed that plays evenly against strong, adaptive poker-bots, plays profitably against non-adaptive poker-bots and against real opponents for play money. Two separate versions of CASPER were tested and the addition of extra cases to the case-base was shown to result in improvements in overall performance. It is interesting to note that CASPER was initially unprofitable against the non-adaptive, aggressive poker-bots. One possible reason for this is that as CASPER was trained using data from players at the adaptive table it perhaps makes sense that they would play evenly, whereas players at the non-adaptive table tend to play much more loosely and aggressively. This means that while CASPER has extensive knowledge about the type of scenarios that often occur at the advanced table, this knowledge is weaker at the non-adaptive table as CASPER runs into situations which it is not familiar with.

Why then was CASPER not profitable on the real money tables. Two hypotheses could explain this. First, that people play poker very differently for real money than for play money. Since CASPER's case-base is derived from poker-bots that are playing for play money these cases are not representative of real money games. The average similarity of cases retrieved when playing against the bots is approx 90%, The average similarity of cases retrieved against real money players is approx. 50%. This is confirmation that CASPER is struggling to find good cases to retrieve, therefore it's poor performance is not a surprise. Our new study has shown that CASPER can improve its play against real people by acquiring a case-base of hands played against real people. We have not the funds to use this case-base against real people for real money. However, we should not assume that the case-base of play money hands would actually improve CASPER's performance for real money games. Indeed, it appears from Figure 4 that CASPER has learned to play more recklessly with play money and that this experience may be disastrous if used with real money.

A second hypothesis we should not discount is that the real money poker games are perhaps in some way dishonest. Many of these websites are run by organizations operating from countries with poor rule of law and little or no legislation concerning online gambling. Ask yourself a question would you play poker online with your money? The first hypothesis could be tested if someone would bank roll CASPER to play 15,000 to 20,000 games online with real money enabling it to acquire a large case-base of real money games that it could then use to guide its play.

It is worth summarizing here our motivation in creating CASPER. We wanted to see if it was possible to create a competitive poker bot that used a simple case-based resoning technique and little or no knowledge engineering. Rather than spending thousand of hours eliciting knowledge from poker experts or creating complex game theoretic algorithms we would simply obtain a library of past poker games and use these alone to inform betting decisions. We believe that the usefulness of CBR has been proven. CASPER plays

competitively against the bots that provided its case-base. Moreover, this case base is competitive against real people for play money since they seem to be playing recklessly. There would seem to be no good reason why if we can obtain a case library from real people playing for real money why our memory based approach would not work. It is our intention to continue this research and compete in a AAAI Poker Bot Competition once we have obtained a suitable case library.

REFERENCES

[1]   Schaeffer, J., J. Culberson, et al. (1992). "A world championship caliber checkers program." Artificial Intelligence 53(2-3): 273-289.

[2]   Schaeffer, J., D. Billings, et al. (1999). "Learning to play strong poker." Proceedings of the ICML-99 Workshop on Machine Learning in Game Playing.

[3]   Billings, D., A. Davidson, et al. (2002). "The challenge of poker." Artificial Intelligence 134 (1-2): 201-240.

[4]   Billings, D., L. Peña, et al. (1999). "Using probabilistic knowledge and simulation to play poker." Proceedings of the sixteenth national conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference: 697-703.

[5]   Billings, D., N. Burch, et al. (2003). "Approximating game-theoretic optimal strategies for full-scale poker." IJCAI.

[6]   Sklansky, D. (1994). "The Theory of Poker." Two Plus Two Publishing Las Vegas, NV, 1994 3rd ed.

[7]   Sklansky, D. and M. Malmuth (1994). "Hold'em Poker for Advanced Players." Two Plus Two Publishing, Las Vegas, NV, 2nd ed.

[8]   Korb, K. B., A. E. Nicholson, et al. (1999). "Bayesian poker." UAI'99 - Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence, Sweden: 343-350.

[9]   Dahl, F. A. (2001). A Reinforcement Learning Algorithm Applied to Simplified Two-Player Texas Hold'em Poker. Proceedings of the 12th European Conference on Machine Learning Springer-Verlag.

[10]   Sandven, A. and B. Tessem (2006). A Case-Based Learner for Poker. The Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006), Helsinki, Finland.

[11]   Salim, M. and P. Rohwer (2005). Poker Opponent Modeling. Indiana University: Personal communication.

[12]   Davidson, A. (2002). Opponent modeling in poker: Learning and acting in a hostile and uncertain environment. Master's thesis, University of Alberta.

[13]   Rubin, J. & Watson, I. (2007). Investigating the Effectiveness of Applying Case-Based Reasoning to the game of Texas Hold'em. In, Proc. of the 20th. Florida Artificial Intelligence Research Society Conference (FLAIRS), Key West, Florida, May 2007. AAAI Press.

[14]   Watson, 1. (1997). Applying Case-Based Reasoning: techniques for enterprise systems. Morgan Kaufmann Publishers Inc. San Francisco, US.