A Case-Based Learner for Poker

Arild Sandven[*] and Bjørnar Tessem[†]
Dept. of Information Science and Media Studies
University of Bergen, Norway
[*]arild.sandven@student.uib.no, [†]bjornar.tessem@uib.no

Abstract

Games have always been an important area for artificial intelligence research, and the topic has become increasingly more important during the later years due to the complexity of today's commercial computer games involving role playing adding psychological elements to the games. The poker card game constitutes a type of game where players have to handle not only statistical uncertainty, but also psychological aspects, which among humans usually are handled less by general statistical principles. Instead we handle psychological aspects by references to previously experienced situations. Thus, in this paper we will describe an approach to case-based reasoning as a technique to improve poker playing ability, as an alternative to rule-based and pure statistical approaches. Results show that the poker playing program is able to learn to play poker at the same level as rule-based systems even with a simple case-based reasoning approach.

1 Introduction

Games have always been a challenging domain for artificial intelligence (AI) research, from the start focusing on zero-sum, deterministic, perfect information games. Today computer games have turned into a big industry with an emphasis on role playing meaning that artificial intelligence components also has to handle psychological aspects of a game, observing behavior, interpreting, and suggest actions based on the interpretation of the situation [16].

The poker card game may on the surface seem to be a game almost of the classic type, but has important characteristics that also makes it a psychological game, involving both interpreting other players behavior and modifying own behavior as the game goes on. This is of course due to the lack of complete information inherit in the game, which makes the player dependent on probability assessments throughout the game. Any information available, including opponent behavior, will be used to assess the situation and decide play. Hence, unpredictable and confusing behavior is considered a winning strategy for poker players. For this reason we have seen several attempts to address poker playing as an AI research topic [3,4,6,9,11,13] many of them with a focus on opponent modeling and psychological aspects of poker [5,11].

It is accepted that humans handle many problem solving situations by referring to previous experiences in the form of cases or situations of similar type. Assessing situations in poker games is also of this kind, both regarding statistical and psychological factors involved. Within AI, Case-Based Reasoning (CBR) [12] is the technique used to perform this kind of problem solving. CBR has not to a great extent been used in game playing but has lately been taken up, for instance in [18] and also at an ICCBR workshop in 2005[1].

In this paper we will describe how a simple CBR tool (FreeCBR) may enable learning in the poker variant Texas Holdem, and we will document this by measuring the learning ability of a case-based poker playing program. We proceed the presentation by giving some background on the poker card game, poker playing software, and CBR in computer games in the next section, describe our *bot* (short for robot, term commonly used for artificial players in computer games) in section three, describe experiments and results in section four, and discuss the results in section five before we conclude.

2 Background

Important motivations for this research are the particular properties of poker which make the game an interesting game of study in AI, as well as the properties of CBR, which makes the approach ap-

[1] http://www.cs.indiana.edu/~davwils/iccbr05games/game-sim-ws.html

plicable to game playing in general, and in particular in poker.

2.1 The game of poker

Poker is a card game played among two and up to about ten players depending on the variant played. Texas Holdem, which is the variant used in our project, allows for 10 players. The goal of the game is to, based on the strength of your cards, outbet your opponents during several rounds of betting. When the last round of betting ends, the player with the strongest five card combination (pair, three of a kind, flush, straight, full house, and other combinations) wins the whole pot, which consists of the complete bets from all participating players. In each round players take turns betting, and are at any time free to withdraw from the game (fold), add an amount of money needed to equal the total highest bet so far in the game(call), or add more money than needed to equal the highest bet (raise). If all remaining players have called, the play continues to the next stage of play, which in Texas Holdem consists of showing more cards to the players. In Texas Holdem it is played four stages, the first with two private cards each, the next with added three common cards, the third with a fourth common card added, and the last with the fifth common card added. The combination for each player is based on the best five card combination of the seven available cards. A fourth kind of bet (check = a bet of 0) is allowed if nobody has bid anything in the stage so far.

High achievement in this game is of course dependent on ability to assess probabilities of winning given the available information in the cards dealt and the bets made so far. It has been developed highly sophisticated models for handling these probabilities given the known cards. However, as players we should also be informed by the bets given by other players, as these may indicate strong or weak hands on their part. This again opens for aggressive and passive playing styles, indicating willingness to bet more or less than the real value of any particular hand, and also bluffs, which are tactical bets on weak hands to make players with stronger hands than your own fold. Thus, achievement in poker also depends on ability to as often as possible to use psychological cues to correct our statistics based assessment of own hand strength compared to others.

In the particular version of Texas Holdem played by our software all raises have to be a fixed amount, and at any stage nobody should need to put in more than five times the raise, (i.e., only five raises are allowed in a round). In addition the game is played with *small blind* and *big blind* roles indicating that in the first stage the first player (small blind) is forced to bid one raise, and the second player (big blind) is forced to raise that bid.

2.2 Poker-playing software

Texas Hold'em has been the subject of extensive research from a group of researchers from the University of Alberta (UoA) over the past decade[2]. They have been able to create a program which is widely regarded as the strongest poker-AI in the world. Their prototypes, Loki and Poki have achieved great success against human opponents, and work is ongoing. They have developed and applied AI-techniques such as neural networks to be used by Loki and Poki. Also, a poker library, Meerkat, which contains useful poker tools, is made available to the public. As of late, the UoA research group have gone commercial, releasing their own Poker Academy™ (BioTools[3]). Lately, they have also widened their support to different types of poker, and they continue to support plug-in bots for amateur programmers.

In addition to the contributions from UoA and BioTools, several other researchers have explored poker from different angles. A simplified version of artificial poker and machine cognition was created as early as 1977 [9]. In the last decade, attempts have been made with poker based on Bayesian networks [13], evolutionary algorithms [3], and adaptive learning [11]. However, none of these have achieved the same attention or success as the UoA projects.

2.3 Case-based reasoning in computer games

Case-based reasoning [12] and its focus on single situational experiences as a problem solving tool provides a flexible and promising approach on domains with "*poorly understood problem areas with complex structured data that changes slowly with time and justification is required*" (Watson, [21]). The process of CBR is cyclical and consists of essentially four steps: retrieve, reuse, revise, and retain. Retrieve involves finding similar cases to a current case, reuse is to adapt and suggest a solution to the current situation, revise is to improve the solution based on other knowledge, and retain is to remember the current case and its solution by adding it to the case base. After each cycle the case base has been extended, and with even more knowledge hopefully has a better chance of providing good solutions to new similar problems. More learning oriented approaches to case-based reasoning is also denoted by the term case-based learning, also

[2] http://www.cs.ualberta.ca/~games

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

often used synonymously with example-based learning or instance-based learning [14].

CBR does not have a long history for use in computer gaming. An early attempt was DeJong and Schulz's Othello playing software [7]. Later approaches include Fagan and Cunningham's work on prediction of human behavior playing Space Invaders [8], Powell et al. [17] who developed a CBR checkers player, Sánchez-Pelegrín and Díaz-Agudo [18] who describe the use of CBR on tactical decisions in a strategy game (C-evo), and Molineux et al. [15] and Aha et al. [2] who describe the use of CBR in different aspects of the real time strategy game Wargus.

3 The Case-Based Poker Player

The artificial player developed in this project was developed in Java and has the name Casey[4]. More precisely, Casey is a special type of client, which connects to a server and responds automatically to events. Contrary to normal poker clients, the human interaction and logic is replaced by a program which decides bids to be given in a game played on the server. Casey is implemented in accordance with the player interface developed at the University of Alberta. This ensures portability between bots to and from present and future poker frameworks. When Casey acts, he starts by sending the necessary game features to the CBR system. The CBR system then creates a target case, performs a search and returns the selected strategy based on Casey's preferences. The functionality of Casey's CBR system is described in the following.

3.1 The CBR-system

FreeCBR is a freeware CBR-system under the Public Domain-license. It offers interaction through a graphical user-interface, command-line, active-X or through Java libraries.

Although it is a quite simple system, FreeCBR proved sufficient for this project. However, research on different implementations of the CBR-system may be fruitful in order to maximize the results of Casey. If maximizing profit is the goal, FreeCBR may not be sufficient, but it is extensive enough to discover whether CBR has potential in this context.

With FreeCBR we may determine similarity using regular data types like String, MultiString, Float, Integer and Bool. The process of determining closest match can be found in the FreeCBR documentation[5]. The closest match is calculated using weighted Euclidian distance. First the case distance is computed from

$$d = \sqrt{\sum_{i=1}^{n} w_i \cdot d_i} \qquad (0)$$

where d_i and w_i are the measured distance and the weight for case feature i respectively. Then the similarity s between two cases is given by

$$s = 100 \cdot (1 - d) \Big/ \sqrt{\sum_{i=1}^{n} w_i} \qquad (0)$$

d_i is as mentioned the distance between the searched feature and the actual case feature. This value is a real number in the interval [0, 1] where 0 means exact hit and 1 means maximum distance. w_i is an integer >= 0, by default set to 5.

In addition, FreeCBR offers possibilities for more selecting alternative strategies for assessing similarity. In our solution we have used the default strategy, which in the documentation is named fuzzy linear and returns a similarity 1 if all features are equal or else the similarity given in equation 2.

3.2 Implementing the CBR-system

In Texas Holdem, one of the most important decisions is made once the player is dealt the first two cards. This stage is known as the *preflop* stage. The decisions of this stage are based on somewhat different features than in the other steges. Thus, we decided to separate the data into two case-bases, the *preflop* and *postflop* base.

Each case consists of two types of features. First, the indexed features are used for retrieval and should be predictive. In the context of poker we have chosen the following features

- *Hand strength (Hnd)*. This is a relative numeric value calculated with the Meerkat Library. This is relative to the number of opponents in the hand. The values range from 0-100 where 100 indicates an unbeatable hand.
- *Relative Position (RPos)*. A relative value to describe the players position. The value ranges from 0.0 to 1.0 where 1.0 means the player acts last, which is considered the best position.
- *Number of opponents (#Opp)*. Indicates how many players are still in the hand. More precisely, *#Opp* is the sum of players who have already called a bet, and the players yet to act.
- *Bets to call(BTC)*. If a player is in the big blind position and no-one has raised the pot, the value is 0, otherwise it would always be positive. If

[4] He should not be confused with CASEY (Kolodner, 1993).

[5] http://freecbr.sourceforge.net

the maximum number of raises is reached before the player has acted yet, the value may be 5.

All these features are highlighted by Sklansky and Malmuth [20], which is regarded as the leading authority on limit poker. If we recall the definition of predictive indices [12], they should be responsible for the way the case was solved, and influence the outcome. We argue that these features combined do just that. First of all, these features dictate how you play your hand because the end result is very much dependant on these features. Also, indices are abstract enough to be widened for future use. Since no specific rules or properties are modeled, these concepts apply to every kind of poker game, not only Texas Holdem. Finally, all indexes are numerical, and on an interval-scale, making them concrete and comparable.

Another type of features in the preflop case-base is the un-indexed features. They are not used in retrieval, but contain valuable contextual information. These indexes are

- *Strategy (Str).* Indicates which strategy is chosen for this scenario. Described in section 3.3.
- *Stage investment(SInv).* This is used to measure the possible negative consequences of a certain strategy. If a player has entered much money in the pot and lost, this will count as a very negative case.
- *Stage pot (Spot).* This is a measure of how much the other players put in the pot. Useful for deciding the most profitable strategy. If this value is high, and the player has won, this will count as a very positive case.
- *Result (Res).* The overall result of all actions in all stages combined. A positive number means the player won, otherwise, the game was lost.

These features can be used to maximize winnings, or minimize losses as they describe the success or failure of a specific case. Stage investment and stage pot describe the risk and reward factors of this particular stage, while result describes the net result of the complete game.

The postflop case base shares many of the features of the preflop case base. However, some differences exist. First of all, *Number of opponents* is no longer recorded. First of all, this was done because hand strength is reduced relative to the number of opponents. Secondly, Casey has information on previous raises (*bets to call*), giving a clear indication on whether the hand is good enough. Thirdly, and most importantly, with decent *hand strength* and several opponents, the pot odds (expected win divided by amount bid so far) increase with several opponents. A hand with no or little chance of winning will be folded regardless of the number of opponents, unless a bluff is likely to be successful.

Some features were also added. First of all, the potential features. The *positive potential* feature calculates the probability of improving the hand. This is very useful for strong *flush draws* and *straight draws*. The *negative potential* calculates the probability of getting outdrawn if the current hand is currently best. This means a high negative potential should induce betting and raising to make drawing opponents pay the maximum price to get lucky. Also, the *potsize* is included, as a big pot would make folding less attractive if you have any kind of *draw*.

Finally, the *relative position* feature is slightly altered. This feature now represents the position relative to the last person to act on this round (if someone bet in first position and was raised, he is now last to act). A raise immediately before your turn is not considered good. Below, we have summarized all features of the postflop case base.

- *Stage (Stg).* Indicates what stage this situation occurred in. 1 = Flop (Board cards 1-3) 2 = Turn (Board card 4) 3 = River (Board card 5)
- *Hand strength (Hnd).* As in preflop
- *Relative position (RPos).* A relative value to describe the players position. The value ranges from 0.0 to 1.0 where 1.0 means the player acts last, which is considered the best position. Note that if a player raises, the player before (if not folded) will be last to act, as he must close the betting.
- *Bets to call (BTC).* Indicates the number of bets and raises in front of the player. 0 means no-one has bet, 5 means betting is capped.
- *Positive potential (PPot).* Indicates how likely it is that the next card will improve your hand.
- *Negative potential (NPot).* Indicates how likely it is that you will lose, given that your hand is currently best.
- *Pot size ($Pot).* How big the pot is right now.

The postflop case base uses the same unindexed features as the preflop base.

After the features have been selected, reasonable weights for measuring overall distance needs to be made. Since we had no prior experience regarding these weights we chose default values wherever possible. This means that all indexed features were given the same weight. During initial testing, some experiments were made with differentiated weights, with especially priority on hand strength. This was later discarded since no immediate effect was observed.

3.3 Implementing strategies

In order to maximize profits in poker, players need to be unpredictable and tricky. To ensure that Casey exhibits these skills, a set of strategies has been developed. There are three available actions for the poker playing clients.

First, there is an aggressive action. This consists of BET/RAISE. More specifically, if nobody has bet, the action will be interpreted as a BET. If there is already a bet in the pot, an aggressive action constitutes a RAISE. Secondly, the passive action ensures that a player never folds, but does not enter more money into the pot than necessary to stay in. Passive actions consist of CHECK/CALL. A CHECK is performed if nobody has opened the betting, otherwise a CALL is performed. Finally, a FOLD action is available.

Note that check equals call, and bet equals raise when a player/bot performs an action. The current game state dictates which action is actually performed. All basic actions are summarized in table 1.

Action*	**Style**	**Description**
FOLD	SURRENDER	Fold
CHECK = CALL	PASSIVE	Check if no bet is made, call if bet is already made
BET= RAISE	AGGRESS-IVE	Bet if no bet is made, raise if bet is already made

Table 1: Player actions available.

To be able to perform advanced strategies, a *Strategy* feature was developed. We developed an algorithm deciding on a suitable strategy for any given game context using the case-bases. A *Strategy* consists of an initial action, and a follow-up response if applicable (depending on whether game-conditions permits or requires a second action from a player in a given stage). *Strategies* can also be categorized as honest and/or deceptive, and also as quit, steal or play. Table 2 gives an overview of strategies, or combinations of actions, that are implemented.

In playing we also used a random strategy option, particularly during training. For instance, if Casey is not yet familiar with a situation, a randomly selected non-fold strategy is chosen. This is done to build a decent sized case-base.

STRATEGY	TYPE	STYLE	GROUP	DESCRIPTION
FOLD	HONEST	SUR-RENDER	FOLD	Fold, check if no bet is made*
BLUFF	DE-CEPT-IVE	AGGRESS-IVE/SUR-RENDER	BLUFF	Bet/raise, fold if re-raised
CALL ANY	BOTH**	PASSIVE	PLAY	Check, call if bet is made
SEMI-BLUFF	DE-CEPT-IVE	AGGRESS-IVE/PASS-IVE	PLAY	Bet/raise, call if re-raised
RAISE ANY	HONEST	AGGRESS-IVE	PLAY	Bet/raise, re-raise if re-raised
CHECK RAISE	DE-CEPT-IVE	PASSIVE/ AGGRESS-IVE	PLAY	Check/call, re-raise again if re-raised

*If no bet is made, it is not smart to fold, a check will give more information at no cost. ** Context-dependent

Table 2: Strategies implemented

3.4 The decision-making process

In poker, several factors must be considered whenever an action is required. Great players balance all these factors, and choose an action to the best of their knowledge. The overall mantra, or heuristics, can be summarized into minimize your losses, but maximize your winnings. Casey uses the case-base when faced with a decision. Whenever a request for action is made from the dealer, a suitable strategy must be chosen based on the experiences recorded in the case base.

Initially, several options were considered. An optimistic best-match search was implemented and tested at first. This returned the strategy chosen from the closest matching case with a non-negative result. This resulted in several strange decisions as several cases were positive because of an incredible amount of luck (such as hitting two perfect cards in a row for a miracle straight). This was obviously not a winning play in the long run, and the algorithm was modified to take a larger number of similar cases into account before making a decision. This way, the effect of extraordinary results was minimized, and safer and more generally applicable strategies were favored instead.

In table 3 we see and example of a summary of previous similar experiences as presented to Casey. This is calculated from all closest matching cases within the given parameters, and is the basis of Casey's decisions. In other words, Casey is presented with a summary of the selected cases. We use the term *generalized case* for this summary, inspired by [1]. The table contains information of how often a game is won under similar circumstances, and the average results for each strategy.

The decision algorithm use information from the generalized case and mainly checks if the case base has enough similar cases, runs a random play if not, else checks whether there is a profitable strategy and plays this. If there is no profitable strategy, it checks whether this is a potential bluff situation and if so, runs a bluff.

STRATEGY	RESULT	FREQUENCY	WINS	LOSSES
Check-Fold	0	2	1	1
Bluff	-20	8	4	4
Call-Any	-35	1	0	1
Semi-Bluff	45	14	8	6
Raise-Any	120	15	7	8
Check-Raise	20	8	5	3
SUM Play	*150*	*38*	*20*	*18*
SUM Total	*130*	*48*	*25*	*23*

Table 3: The generalized case

Note that before we had enough cases matching the required similarity threshold we let Casey play a random strategy. This we did in order to fill the case base and get experience with different strategies in all situations. In the experiments, after 10.000 hands, the most common situations were familiar, and after about 20.000 hands random strategies were hardly used.

We also had to handle bluffs differently. We experienced that if a bluff was attempted and called, and it still led to a win for the bluffer, this would count as a good case, when in reality it meant that the bluff was unsuccessful. On the opposite side, if a bluff was randomly executed with great cards, this could lead to a fold if was re-raised (A bluff consists of BET-FOLD). We therefore decided to separate actions and strategies into three groups, PLAY, BLUFF or FOLD. The algorithm for selecting the best strategy from the generalized case thus only compares strategies in the PLAY group, and bluffs are chosen randomly only if no winning strategy is found.

After selecting a strategy according to these heuristics, the case is temporarily stored. At the end of each game the cases of that game is updated with information about the final result. The updated cases are then stored in the case base, and his knowledge base continues to grow.

4 Experiments and Results

The learning experiments for Casey consisted on runs of 50.000 poker hands. This took one week to run on this system, which of course limited the number of experiments to run. We ran he experiments with a similarity threshold of 95, which meant that if the case base did not have enough cases with higher similarity for the generalized case, Casey would choose a random strategy. Finally we used a maximum of 100 cases, i.e., the 100 most similar cases in a generalized case. Also decisions for case was based on the stage results rather than final results for the game, which is according to recommended poker playing behavior.

The first initial tests of Casey showed promise. He was run against a random player with absolutely no rules or learning abilities. As expected, Casey started winning almost immediately, needing only a few hundred hands to learn how to beat this opponent consistently. What he learned was to be aggressive on almost every hand, causing the random player to fold every 6th action (6 strategies to choose from), regardless of what hand he held.

For the main experiments, Casey played against instances of *RuleBot*, which is a not very sophisticated bot provided in the Meerkat library. However, the rules implemented suggest a safe and sound form of poker. It is not adaptive and without opponent modeling, but as opponent modeling is yet to be introduced in Casey, the bots compete on equal terms, treating each game as a game against unknown players.

The number of opponents should have an effect on your playing style. In a 2-man game, you figure to have the best hand every other hand, and should therefore play almost every other hand. In an 8-man game, such a strategy would probably be disastrous. We decided to test Casey against 3, 5 and 7 opponents to check his abilities with different types of games, and whether he was adaptable to such changes. Before changing the number of opponents the case base was reset, requiring Casey to start learning from scratch each time.

The success of Casey is dependant on two main factors. First, his results which are measured on a simple numeric scale. Second, his learning ability which is indicated by variation in results measured over time. Also, it is interesting whether he is able to mimic play related to other ideas, like the importance of position, or to protect your blinds. These abilities are not as important as the results and progress, but could give an indication of strengths and weaknesses by applying CBR in a game of poker. By investigating the database some of these questions may be answered, and we will return to this in the discussion.

Anyhow, two variables were measured and examined. First, the actual *result*, or how much Casey has won or lost totally at any time (balance would be an alternative concept). This gives an indication of his level of play. Secondly, we looked at the *fluctuations*, which is a measure of the difference between his highest and lowest result over a sequence of games. This indicates whether the results are produced by random luck (high fluctuation), but also it gives us an indication as to when his initial learning phase is over (when the fluctuations are fairly stable from phase to phase). We divided the 50.000 hands into 5 phases, each with 10.000 hands, and measured fluctuation within each of these phases.

Figures 1 and 2 give an overview of how results varied during the experiment runs and fluctuations in the five phases.

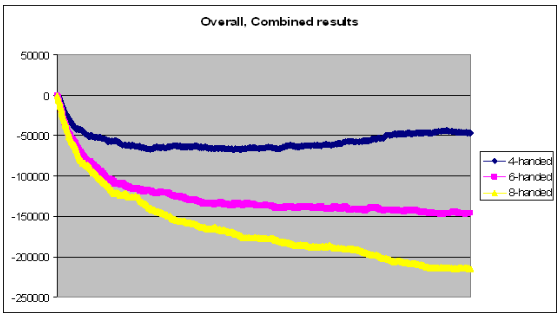

Figure 1: Overall results

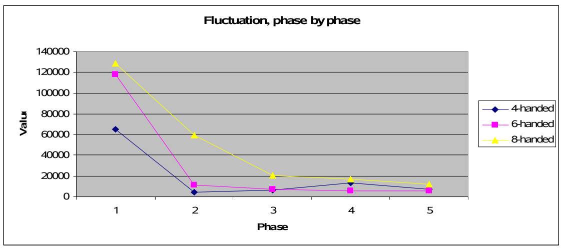

Figure 2: Summary of fluctuations

In the first phase, which spans from hand 0 to hand 10.000, Casey starts with absolutely no knowledge at all due to mainly random play, and to gain experience, he never folds. Naturally, playing every hand will lead to a negative result, as indicated in the figure above. As expected, the curve is steeper with 6 or 8 players as a wider variety of situations are encountered.

The second phase shows signs of promise in the games with fewer players. 4-handed, Casey already plays on level with RuleBot. 6-handed also shows significant improvement, both in terms of results and fluctuation, indicating that the initial learning phase is over. The 8-handed results are still far from satisfactory, but the curves indicate progress.

In the third phase, all simulation results seem to stabilize. Once more, Casey still is more successful in shorthanded (few opponents) games, showing a profit for the first time in the 4-handed game. 6-handed and 8-handed he is still losing, but once more showing progress from the previous phase. Fluctuations are still decreasing, an indication that he has not yet reached his peak.

In phase 4 Casey is still improving, but just slightly, and the results are getting more consistent. The 4-handed game is producing a nice profit, while 6-handed are almost on par, reproducing the indications from the previous phase. However, he is still trailing in the 8-handed game, which indicates that additional work is needed in order to produce a positive result here. It is also worth noting that the fluctuation of the 4-handed game has increased. This is a positive fluctuation, produced by winning play, and could indicate that Casey has bigger potential shorthanded, not yet reaching his peak. But he is still losing in the other games, and although not much, the fluctuations are not reduced significantly.

The last phase in the simulations presents some interesting results. In the 4-handed game, Casey still has a positive result. However, his average result is significantly reduced. If we look at the graph, this seems to stem from the last 3000 hands. This could indicate a natural run of bad cards and bad luck, or an especially lucky streak in the previous phase. Another possible explanation is that he has learned more about starting hands, being more selectively aggressive, and playing fewer hands. However, if we disregard the last 3000 hands, his average is on par with the results from previous phase (steady progress from 30.000 to 47.000), nurturing the assumption that he has experienced bad cards and bad luck. In the 6-handed game, the results are essentially the same. This indicates that the CBR system has taken Casey as far as it can. The fluctuations also stay the same, suggesting that further simulations would produce no significant progress or regress. Still, he plays almost on par with RuleBot, quite an accomplishment in our opinion. In the 8-handed game, fluctuation is once again reduced. Clearer indications would require at least 50.000 additional hands, as he probably has not reached his peak quite yet. His results are also still improving. However, since he seems to have reached his peak in the 4-handed game, we believe that additional test for the 8-handed game would only produce results approaching that of the 6-handed game.

5 Discussion

Casey did learn during experiments, but was unable to become a winning player against many opponents. However, playing fewer opponents, he performs very well. All-in-all, he performs reasonably well compared to the rule-based bots. We have seen that Casey clearly performs better when faced with fewer opponents. The probable explanation for this is his aggressive style during training, which is essential in shorthanded play [10, 19]. However, when opponents are many, much better cards are needed to win. Although this is picked up during play, the result seems to be that Casey himself becomes timid, folding marginal and great drawing hands. In other words, Casey learns hand strength to some degree, but not hand potential. On the other side, Casey seems to be able to exploit opponents betting patterns, by betting on weakness and folding when opponents show strength. Below we discuss some abilities that are considered desirable for good play and our assessment of Casey's performance regarding those abilities by using the plays database.

Starting hand requirements – Does he play the right hands?
One of the most important qualities of a winning player is playing the starting hands better than your opponents. Since good starting hands end up as winners more often, this gives the player a mathematical advantage. A good indication of this judgement is given by the percentage number of hands won when a hand is played. If we consider the 8-hand table, we can see that he wins only 6337 out of 15578 hands played in the flop stage, or about 40%. This indicates that his starting hands requirements need more work, as he clearly plays to many hands with little chance of winning.

Adaptability - Does he adapt to different number of opponents?
If we take a look at the percentage number of hands played in the three different simulations, we discover that his percentage drops with increased opposition. 4-handed, he plays about 37% of the hands. This percentage stays at 37% at the 6-handed table, partly because of the prolonged learning period necessary when facing more opponents, and therefore a broader variety of situations. However, 8-handed, his participation is down to 31%, with an even longer learning period. So, Casey experiences that it is difficult to win against more opponents, and plays fewer hands.

Pot odds concept – Can he identify valuable plays even with weaker hands?
From observing Casey, this does not seem to be the case. He throws away strong drawing hands from time to time, and occasionally also folds with *overpairs* (pair on hand with card value higher than any of the common cards). These problems are not easily corrected without implementing rules for pot odds and potential.

Bluffing – Is he able to exploit signs of weakness in his opponents?
Casey clearly exploits opponent weakness. he is aggressive in 4-handed games, and reduce his aggressiveness as the opposition increases. This is not merely a sign of pure bluffing skills, but also another indication that he adopts suitable playing styles according to changing table conditions.

Positional sense – Does he play more hands in late position?
If we look at the relative number of hands played in first vs. last position, Casey clearly prefers the latter. In the 8-handed game, Casey played 872 out of 6264 possible hands preflop from early position, or about 13% (first to act, position 2). When on the button (last to act, position 7), his participation was increased to 2187 out of 6203, or about 35% of the possible games. Playing more hands in late positions are considered good practice.

Blind play – Does he protect his blinds?
Having already highlighted the importance of positioning and exploiting weakness, it is also important to stop your opponents from doing the same to you. This is especially common against the blinds, as they have made forced bets and are likely to hold bad or mediocre hands. That means that it is necessary to protect your blinds by playing an increased number of hands that you usually would fold. Casey plays more blind hands than any of his opponents. Still, it is not easy to see whether Casey protects his blinds well or even over-protects them, making this a losing play. It could also indicate too low starting hand requirements, which is perhaps the most common source of losing.

Showdown – Does he know when to let bad hands go?
One last and perhaps most important ability is to let your bad hands go at the earliest possible moment. This is often measured by the percentage of *showdowns* (the deciding moment of a play when all the remaining players show their cards to decide who wins the pot) won. While showing down the best hand makes you money, it also creates an impression that you only play strong hands, setting up profitable bluffing opportunities later on. On the 8-hand table Casey wins, or breaks even 6337 times

out of 7488 showdowns, or about 85%. This is a strong result, and indicates an ability to throw away losing cards.

6 Conclusion

In short, our experiments indicate that a CBR system for poker play is able to develop playing strength and some recommended styles of play. Still, we see that some abilities are not learned satisfactory, partly due to lack of opponent models. Good opponent models would be able to quickly categorize opponents and use this information as an additional feature when selecting cases. In fact, CBR may indeed be of most value in opponent modelling. In addition we see a weakness in Casey's ability to consider pot odds and in assessing the start hands. Both opponent modelling, pot odds, and start hand assessment may be the focus of future work.

References

1. Aamodt, A. and E. Plaza (1994). "Case-based reasoning: foundational issues, methodological variations, and system approaches." AI Commun. **7**(1): 39-59.
2. Aha, D.W., Molineaux, M., and Ponsen, M. (2005). Learning to win: case-based plan selection in a real-time strategy game. Proceedings of the Sixth International Conference on Case-Based Reasoning (pp. 15-20). Chicago, IL: Springer.
3. Barone L. and While L. An Adaptive Learning Model for Simplified Poker Using Evolutionary Algorithms. In proceedings of Congress of Evolutionary Computation 1999 (CEC'99), July 6-9, Washington DC, pp 153-160, 1999.
4. Billings, D., Papp, D., Schaeffer, J., and Szafron, D. 1998. Poker as Testbed for AI Research. In *Proceedings of the 12th Biennial Conference of the Canadian Society For Computational Studies of intelligence on Advances in Artificial intelligence* (June 18 - 20, 1998). R. E. Mercer and E. Neufeld, Eds. Lecture Notes In Computer Science, vol. 1418. Springer-Verlag, London, 228-238.
5. Billings, D., Papp, D., Schaeffer, J., and Szafron, D. 1998. Opponent modeling in poker. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial intelligence/innovative Applications of Artificial intelligence* (Madison, Wisconsin, United States). American Association for Artificial Intelligence, Menlo Park, CA, 493-499.
6. Billings, D., Davidson, A., Schaeffer, J., and Szafron, D. 2002. The challenge of poker. *Artif. Intell.* 134, 1-2 (Jan. 2002), 201-240.
7. De Jong, K., & Schultz, A. C. (1988). Using experience-based learning in game playing. Proceedings of the Fifth International Conference on Machine Learning (pp. 284-290). Ann Arbor, MI: Morgan Kaufmann.
8. Fagan, M., & Cunningham, P. (2003). Case-based plan recognition in computer games. Proceedings of the Fifth ICCBR (pp. 161-170). Trondheim, Norway: Springer.
9. Nicholas V. Findler. Studies in machine cognition using the game of poker. Communications of the ACM, 20(4):230--245, April 1977.
10. Jones, L. (2000). Winning low-limit hold'em. ConJelCo. Pittsburgh, Pa.,
11. Kendall, G. and Willdig, M. 2001. An Investigation of an Adaptive Poker Player. In *Proceedings of the 14th Australian Joint Conference on Artificial intelligence: Advances in Artificial intelligence* (December 10 - 14, 2001). M. Stumptner, D. Corbett, and M. J. Brooks, Eds. Lecture Notes In Computer Science, vol. 2256. Springer-Verlag, London, 189-200.
12. Kolodner, J. 1993 *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc.
13. Korb, K. B., Nicholson, A. E., and Jitnah, N. 1999. Bayesian poker. In Proc. 15th Conference on Uncertainty in Articial Intelligence (1999). Morgan Kaufman.
14. Mitchell, T. M. 1997 *Machine Learning*. 1st. McGraw-Hill Higher Education.
15. Molineaux, M., Aha, D.W., & Ponsen, M. (2005). Defeating novel opponents in a real-time strategy game. In D.W. Aha, H. Muñoz-Avila, & M. van Lent (Eds.) Reasoning Representation, and Learning in Computer Games: Papers from the IJCAI Workshop (Technical Report AIC-05-127). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.
16. Alexander Nareyek. Artificial Intelligence in Computer Games - State of the Art and Future Directions. ACM Queue 1(10), 58-65, 2004.
17. Powell, J.H., Hauff, B.M., & Hastings, J.D. (2004). Utilizing case-based reasoning and automatic case elicitation to develop a self-taught knowledgeable agent. In D. Fu & J. Orkin (Eds.) Challenges in Game Artificial Intelligence: Papers from the AAAI Workshop (Technical Report WS-04-04). San Jose, CA: AAAI Press.
18. Rubén Sánchez-Pelegrín and Belén Díaz-Agudo. An intelligent decision module based on CBR for C-evo. In David W. Aha, Héctor Muñoz-Avila, & Michael van Lent (Eds.) Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games (http://home.earthlink.net/~dwaha/research/meetings/ijcai05-rrlcgw), Edinburgh, Scotland, 31 July 2005, pp. 90-94.
19. Sklansky, D. (1997). Hold'em poker. Two Plus Two Pub. Hendersen, NV.
20. Sklansky, D. and M. Malmuth (2001). Hold'em poker for advanced players. Henderson, Nev.
21. Watson, I. 1998 Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann Publishers Inc.