

DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS C
REPORT C-2008-41

**A Heuristic Based Approach for a Betting Strategy
in Texas Hold'em Poker**

Teemu Saukonoja and Tomi A. Pasanen

UNIVERSITY OF HELSINKI
FINLAND

A Heuristic Based Approach for a Betting Strategy in Texas Hold'em Poker

Teemu Saukonoja and Tomi A. Pasanen

Department of Computer Science
P.O. Box 68, FIN-00014 University of Helsinki, Finland
Gamics Laboratory

Technical report, Series of Publications C, Report C-2008-41
Helsinki, April 2008, 8 pages

Abstract

Artificial intelligence research in Texas Hold'em poker has recently mainly focused on heads-up fixed limit game. Game theoretic methods used in the poker agents capable of playing at the very best level are not easily generalized to other forms of Texas Hold'em poker. In this paper we present a general approach to build a poker agent, where the betting strategy is defined by estimating the expected values of the actions. This approach suits well larger number of players and it is also easily modified to suite no limit games. We focus on defining of the evaluation function. It has a very crucial part at the overall level of the poker agent in this kind of solution.

Computing Reviews (1998) Categories and Subject Descriptors:

- G.3 Probability and Statistics — probabilistic algorithms
- I.2.1 Artificial Intelligence: Applications and Expert Systems — games
- I.2.8 Artificial Intelligence: Problem Solving, Control Methods, and Search — heuristic methods

General Terms:

Card games, Texas Hold'em poker

Additional Key Words and Phrases:

1 Introduction

The popularity of poker games and especially the popularity of Texas Hold'em poker have significantly grown in the last few years. Nowadays, Texas Hold'em poker is also a very popular testbed for an artificial intelligence research and recently made poker agents are already capable of playing at the world-class level in some forms of Texas Hold'em poker.

Texas Hold'em poker suits well as an artificial intelligence research because of its nature, imperfect information and randomness that cause uncertainty in decision-making. It also provides a well-abstracted field for the study. In Texas Hold'em poker, it is also possible to use very different kinds of methods trying to build an agent capable of succeeding in this field. In this paper we assume that a reader knows Texas Hold'em as a game. If the game is not familiar or for additional information about the game, the problem field and the strategies, we recommend reading Sklansky (1994), which is considered as the bible of the poker.

Lately, the artificial intelligence research in Texas Hold'em poker has mainly focused on fixed limit heads-up game, where the results have been very promising. Methods used in these agents are based on approximation of the game theoretic equilibrium strategy (Johanson, 2007). Even if these methods are very suitable for two player fixed limit Texas Hold'em, there is no guarantee that these methods could be used in any other form of Texas Hold'em poker.

No limit Texas Hold'em poker is strategically much more complex. When betting in poker is limited, the errors made in the game are not as crucial. It is also easy to notice, when adding a new attribute, the betting amount, to the game, the number of states in the game increases exponentially. Even if multiplayer Texas Hold'em is considered strategically less complex than the heads-up game, the grown number of players also exponentially increases the number of the game states. That is why the experimental results in these forms of Texas Hold'em poker are not as good as they are in the fixed limit heads-up game.

Heuristic expert knowledge or simulation based methods have been used in some agents, mainly focused on playing full table fixed limit Texas Hold'em poker. Systems based on these methods, like Poki (Billings et al., 2002), are still the best choice, when building an agent that is capable to play at the decent level in tables having more than two players. In these kind of solutions, if the betting strategy is not based on the simulations, evaluation function is a critical part of the system.

Now we present a method to build a betting strategy in a poker agent by a heuristic approach, where we define the structure of the evaluation function. The approach has some advantages over methods mainly used to build poker agents. Firstly, it is a very simple and flexible so it can be used to build poker agents capable to play poker at the different levels. Secondly, it is easy to generalize into different amounts of players and even into the no limit games. The approach needs some minor modifications to being able to play at its best possible level in these other forms of the game, but the framework still remain the same.

2 Defining an Evaluation Function in Poker

To build a poker agent based on a heuristic betting strategy, we need a way to estimate expected values of the actions. Estimation is done by an evaluation function that will calculate the expected values from the given parameters. These parameters can be derived by opponent modelling system or they can be defined by expert knowledge. The other method used to do the estimation is to use simulations to define the expected values of the actions directly. When building a poker agent, the rest of the structure of the agent can be similar, with these two methods, and they also can be used together (Billings et al., 2002).

A heuristic evaluation function in a poker agent has a role to unify all the information got from an opponent modelling system and from expert knowledge that has used to build an agent. The role that it has is quite different versus classical two-player perfect information games like chess, where the evaluation function is used to estimate the goodness of a game state. In these games, the evaluation function is usually used with the game-tree search, when the search is cut in some level of the tree.

There are many possible ways to build an evaluation function in Texas Hold'em poker. In this paper we present two of these different ways to do that; *strategic method* and *direct estimation*. Strategic method is more expert knowledge derived, when in the direct estimation method, opponent modelling plays a bigger role.

2.1 Strategic Method

In this method, expert knowledge is used to define the possible different scenarios, how the particular hand could end up. The scenarios can be very accurate. For example, there can be at least couple of scenarios in a situation where the agent bluffs by doing a check-raise on the flop. In this kind of situation, there are a lot of possible alternative endings and these all can have an own scenario. If the scenarios are very accurate, there can be tens or hundreds or different scenarios to build. It is also possible to have only few different scenarios, if we group the situations into different scenarios only if the outcome is highly different in these situations. In this kind of situation the defined five scenarios can be for example:

- Plays, where the player first bets/raises and continues to bet/raise or call to the end.
- Plays, where the player bets/raises and eventually all the opponents fold their hands.
- Plays, where the player bets/raises and one or several of the opponents raise, when the player folds in some point of the hand.
- Plays, where the player checks/calls first and checks/calls or raises later in the hand that ends up to a showdown.
- Plays, where the player checks/calls first and folds later in some point of the hand.

As we see, it is possible to define very different amount of scenarios. More scenarios you have, the easier is the task to derive the estimated pot sizes and amount of bets. The flaw of having a great number of different scenarios is that the overall structure of the evaluation function comes more incoherent and it is more difficult to test or maintain the system. Even if the amount of the different scenarios is not fixed, the expected values of all the possible scenarios can be calculated with only three formulas presented here.

Scenarios that will end at the showdown can be calculated as follows:

$$E(s) = P_w \times ES - EB, \quad (1)$$

where P_w is the agents probability to win the hand, ES is the estimated total pot size and EB is the estimated total amount of bets the agent will invest in this particular hand. Similarly the scenarios that will end to every other player to fold their hands can be calculated:

$$E(s) = P_{vf} \times (ES - EB), \quad (2)$$

where P_{vf} is the probability of the situation, where every other player folds their hands. The scenarios where the poker agent itself folds its hand, can be calculated:

$$E(s) = P_{hf} \times (-EB), \quad (3)$$

where P_h is the probability that the agent itself folds its hand. Now we can calculate the overall expected value of the action as follows:

$$E(a) = \sum_s P(s)E(s), \quad (4)$$

where $P(s)$ is the probability of scenario s to occur.

The defining of the attributes can be done by expert knowledge, opponent modelling or with both of these together. Usually the combination of these two is, by far, the best choice. That how much each of these methods should be used depends on the accuracy of the opponent modelling system. If the agent includes a very developed opponent modelling system, there is no need to use the expert knowledge so much. Instead, if the opponent modelling system is very simple, expert knowledge based functions are needed to make system able to use this opponent modelling data to define the parameters. Choosing the opponent modelling methods and accuracy is handled more later.

In this method if we use a lot of scenarios, the opponent modelling has to be able to give the probabilities of different scenarios or these should be able to calculate by the heuristic functions. With heuristic functions these attributes can be calculated if we e.g. have fixed these probabilities in situations or we use statistical parameters by opponent modelling to define these probabilities.

2.2 Direct Estimation

Direct estimation method is more straight-forward than presented strategic method. Now the ground of the evaluation function is the formula to calculate the overall expected value of the action. As we saw in the strategic method, less scenarios we have, more difficult it is to derive the estimated size of the bets and total pot. In this method it behaves that these attributes are easily derived from the opponent modelling data.

When we unify all the different ending having hands presented in the strategic method, the overall expected value can be calculated:

$$E(t) = (P_w \times ES_w - EB_w) + P_{vf} \times (ES_{vf} - EB_{vf}) - P_{hf} \times EB_{hf}. \quad (5)$$

The needed attributes are the same that in the strategic method, but now we should be able to estimate them directly. There are also different ways to do that. The methods that can be used to define the attributes are pretty much the same as presented with the strategic method, but advanced opponent modelling techniques guarantee the better outcome. We can do that by heuristic functions that are based on expert knowledge or we can use opponent modelling to define these attributes. If we use expert knowledge based functions, the functions can be made to fit against average opponents, but they do not fit very well against an opponent that uses an unorthodox strategy. It also can lead to a situation that opponent has an easier task to exploit the flaws in the game, if we don't use much randomness in our actions.

It is noticeable that even if we are calculating the overall expected value of the action, it is just the overall expected value of the action in the particular hand. There is no guarantee that the overall expected value of the action in the particular hand would be the overall expected value in the whole game. Without noticing this fact we give too much information to the opponent about our hand strength and possible actions.

3 Building a System

When building a system, we have to handle the thing that it is not always the best way to choose the action that maximizes the positive expected value in the current hand. If we can't pay attention to that in the evaluation function, we need another way to deal with that thing. That is why we have to use randomized strategy, where a probability distribution defines the action to choose (Koller and Pfeffer, 1995). In the system we use an action selector to choose the right action based on the randomized strategy and the estimated expected values.

In addition to these methods that actually define the betting strategy, a classical model of building a poker agent includes an opponent modeller and a hand evaluator. Evaluating the strength of our own hand is the most trivial part of the artificial intelligence system. The hand strength can be seen as a probability to win the current hand. It includes the current strength of a hand and the estimated change of the strength in other stages of the game. Opponent modeller can be used e.g. to inference the probability distribution of the opponent's possible hands or actions.

3.1 Opponent Modelling

As we already touched on the last chapter, the opponent modelling plays a big role in this kind of solution. Depending on which method we use as an evaluation function and how expert knowledge independent we would like to be, there are different ways to execute the opponent modelling. An orthodox way to do the opponent modelling is to use some statistical parameters and use expert knowledge with them. Also the system can calculate the probability distribution of the opponent's possible hands, based on opponent's actions, and the probability to win the current hand can be derived from that.

In addition to statistical methods, also other methods can be used to execute opponent modelling. For example neural networks suit well on this task (Davidson et al., 2000). Unifying two or more of these methods is also possible. The different methods can vote for the answer given by the opponent modeller, weighted by the past accuracy of these methods.

3.2 Description of the System

Our experimental system includes the main components as follows:

- Evaluation function.
- Action selector.
- Opponent modeller.
- Hand evaluator.
- Rule-based expert system for the pre-flop betting strategy.

In the post-flop betting strategy, we used the direct estimation technique as an evaluation function. Furthermore, we used an action selector to choose the action probabilistic of the actions, in which the estimated expected value is positive. The action that provides a greater expected value is more likely to be chosen. In addition to that, we used a method that we called as *continued aggression*, where the last betting round started aggression is continued as a greater probability.

We used simple hand strength evaluation to evaluate the effective hand strength EHS that includes the current hand strength HS and the positive $PPOT$ and negative potential $NPOT$ of

the hand with the upcoming card at the next stage of the game presented in Billings et al. (2002):

$$EHS = HS \times (1 - NPOT) + (1 - HS) \times PPOT. \quad (6)$$

The current hand strength is the probability to have the best hand at the moment. The positive potential is the percentage of the hands we are at the moment behind, but expected after the next card ahead. The negative potential is the percentage of the hands we are at the moment ahead, but expected after next card behind.

As an opponent modelling system we used our own two-stage method. Firstly, the systems gathers up some basic information about the action made by the opponents and use these with simple heuristic functions to define the parameters needed by the evaluation function. Alongside with that, the system saves the complete *betting chain* of every hand played in the game. As a betting chain we mean every action made in the current hand chained together. When enough similar chains to the current point of the hand is collected, the parameters needed by the evaluation function are used directly. Below is presented the description of the evaluation function together with the used opponent modelling system:

```
function calcEV(act)

  ch <- current betting chain + act
  Cx <- chains similar to ch
  nx <- number of chains in Cx
  hs <- player's hand strength

  if (nx > set condition)

    Cs <- chains in Cx ending
      showdown
    oh <- average opponent's hand
      strength in Cs
    pw <- calculated probability to
      win by heuristic function
      given hs and oh
    sw <- average pot size in Cs
    bw <- average bets in Cs
    Cv <- chains in Cx ending
      opponent to fold
    nv <- number of chains in Cv
    pv <- nv/nx
    sv <- average pot size in Cv
    bv <- average bets in Cv
    Ch <- chains in Cx ending
      player to fold
    nh <- number of chains in Ch
    ph <- nh/nx
    bh <- average bets in Ch

  else calculate needed parameters
```

```

by heuristic functions

return (pw * sw - bw)
        + pv * (sv - bv) - ph * bh

```

The heuristic functions used in the system are very simple and the values returned are not even meant to be very accurate. They only give an rough approximation used before enough data is collected. Used pre-flop betting strategy is made by a probabilistic rule-based expert system. The total effect of the pre-flop play is so insignificant that there was no need to use advanced methods.

4 Experimental Results

Even if the approach is mainly focused on multiplayer games, we decided to test the approach against several other poker agents in fixed limit heads-up Texas Hold'em. The reason to choose fixed limit heads-up game was that in that form of Texas Hold'em poker, there already are some benchmark programs that are well suitable for testing.

We tested our system against three different systems that all are based on very different kind of methods. Sparbot (Billings et al., 2003) is a game theoretic pseudo-optimal player, Vexbot (Billings et al., 2004) is a learning player, based on game-tree searches and Pokibrat is a heads-up variant of the Poki system. For more information about the test process and methods can be found at Saukonoja (2007).

Table 1: Test results against the opponents.

Opponent	Result (SB/h)
Sparbot	-0.023
Vexbot	-0.242
Pokibrat	+0.015

Results shown in table 1 are only indicative of the real game level of the system. However, it is noticeable that against Sparbot and Poki systems our system is quite competitive. Instead against Vexbot system there can be seen the Vexbot's good adaptation skills, where it can exploit the flaws of the opponent after a period of played hands. In this case the period was about 1000 of hands.

There are some flaws in the betting strategy of the system that are for a learning opponent possible to exploit. Firstly, avoiding exploitation could be done by increasing randomness in the action selector. Secondly, better opponent modelling could lead to better adaptation and to some counter-strategy against the opponent's strategy to exploit the flaws.

There is also possibility to develop the evaluation function and other heuristic functions. Interesting part would be trying to find evaluation function that itself takes into account the whole game, not just the current hand. How that could be done is lot trickier question. Fixing these flaws and develop the system better level is not an easy task to do. Testing and developing the system through exploited flaws is very time-consuming process because of the randomness. In Zinkevich et al. (2006) is presented some methods to help this process.

5 Conclusion

The presented heuristic approach offers a general way to execute the betting strategy in Texas Hold'em poker. The experimental test shows that there are some flaws in the system, but they are not crucial. System fits right away, even better, for multiplayer fixed limit games, where opponents are not able to exploit the flaws of the system as easily as in the heads-up game. Method is also worth of trying in no limit games.

However, we have to remember that if we want to build a system that is capable of playing at the world-class level in no limit Texas Hold'em poker, there are some features that make the task much more challenging; now an agent is able to lost its whole current stack in a one particular hand. Usually the amount can be something like 100 big blinds, but it can be even more. It is noticeable that lost can occur by only one mistake, when in fixed limit game one mistake in a game is not as crucial when you can loose maximum of one big bet, which is only two big blinds. So in no limit game the cost of one mistake can be 50 times the cost of one mistake in fixed limit game. On the other hand, avoiding of errors can lead to an unoptimal play.

The important role of errors in decision-making in no limit games increases the importance of opponent modelling from before. It also makes it possible to use new methods in systems. For example the opponent modelling system can make observations about the reasons that led to errors in decision-making process (Chaddock et al., 2007).

References

- Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.
- Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.
- Darse Billings, Aaron Davidson, Terence Schauenberg, Neil Burch, Michael Bowling, Robert Holte, Jonathan Schaeffer, and Duane Szafron. Game tree search with adaptation in stochastic imperfect information games. In *Proceedings of the Computers and Games: 4th International Conference (CG'04)*, 2004.
- Gabe Chaddock, Marc Pickett, Tom Armstrong, and Tim Oates. Models of strategic deficiency and poker. In *Working Notes of the AAI Workshop on Plan, Activity, and Intent Recognition (PAIR)*, pages 31–36, 2007.
- Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI'2000)*, pages 1467–1473, 2000.
- Michael Johanson. Robust strategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta, October 2007.
- Daphne Koller and Avi Pfeffer. Generating and solving imperfect information games. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1185–1192, Montreal, Canada, August 1995.
- Teemu Saukonoja. Heuristisen panostusstrategian toteuttaminen pokerin tekolysovelluksissa. Master's thesis, University of Helsinki, November 2007.

David Sklansky. *Theory of Poker*. Two Plus Two Publishing, 1994.

M. Zinkevich, M. Bowling, N. Bard, M. Kan, and D. Billings. Optimal unbiased estimators for evaluating agent performance. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, pages 573–578, 2006.