Pseudo-Optimal Solutions to Texas Hold'em Poker with Improved Chance Node Abstraction

Ian Fellows
University of California, San Diego
San Diego, CA
ifellows@ucsd.edu

Abstract

Three game theoretic solutions to two-player limit Texas Hold'em poker are presented. To overcome the problems associated with the large game tree of poker, simplifications and abstractions are employed to reduce the size of the game tree while preserving the probabilistic structure of the game and the strategic information available to the player. This pseudo-poker game is then solved approximately using a co-evolutionary algorithm. Unlike other AI systems in poker, the bucketing abstraction is based empirically on the strategic structure of the game, and public information about the board cards is incorporated into the pseudo-poker game. Our three players INOT, Fell Omen 1 and Fell Omen 2 are evaluated against the work of researchers around the world at the AAAI's computer poker competition. INOT won silver in the 2007 competition, and Fell Omen 2 was in a three-way tie for second the 2008 competition. All three programs have been released as open source projects, and represent the first publically available Texas Hold'em AI systems that play at a near champion level.

1 INTRODUCTION

Texas Hold'em poker is a popular card game of both skill and luck that has become extremely popular over the last 10 years. It is a four round game where players have private cards that only they can see, and shared community cards which all players can see. The progression of the two-player limit variant of the game is as follows:

Anti: The dealer puts 1 bet in the pot (the center of the table), the other player puts in ½ a bet.

Deal: Both players are dealt two cards face down, which they can see but their opponent cannot.

Preflop: A round of betting occurs where each player can raise 1 bet, call or fold. A total of 4 bets are allowed.

Flop: Three cards are put face up in the center (the board). Another round of betting occurs

Turn: One card is put face up in the center. Another betting round occurs, but the amount of each bet increases to 2.

River: A final card is dealt face up in the center and another round of betting is done.

Showdown: If neither player has folded, each player shows his cards. For each player, the best five card hand constructed out of their private cards, and the cards in the center is found. The player with the best five card hand wins.

2 PRIOR WORK

Texas Hold'em poker is a fascinating and challenging game to attack, both because of it's popularity and the difficulty of dealing with its huge game tree, which is O(10^18). Unlike chess or backgammon, poker is a game of imperfect information (players don't know their opponent's cards), because of this, different approaches must be taken, as the usual minimax algorithms don't apply to imperfect information games

The first expert player of two-player texas hold'em was created by Billings et al. (2003). Billing's program psyopti was based on a Nash Equalibrium of a simplified game of poker, which had a drastically reduced game tree (O(10^7)), and thus within the realm of tractability. To solve this smaller game the authors used a linear programming based algorithm on the sequence form of the game (Kroller et al. 1997). This approach produced an advanced player that could for the first time compete with seasoned poker players. The disadvantage was that the algorithm is very memory hungry, and this limited the size of the game that could be solved.

Gilpin and Sandholm (2006) created the bots GS1 and later GS2, which built upon the fundamental ideas pioneered by psyopti, but differed in several interesting ways. GS1 and GS2 like psyopti solved an abstraction of the game of poker, but where psyopti is based on a pre-computed solution to an expert defined game abstraction, GS1/2 use an adaptive computer defined abstraction of the game, and compute parts of the solution in real time. These improvements greatly increased performance, but the size of the abstraction able to be solved was still limited by the linear programming algorithm, and the amount of time available to make decisions in a real time poker game.

The University of Alberta recognized the linear programming limitation, and built their new AI, hyperborean, using a different solution technique (Johnanson, 2007). While the linear programming algorithm yields an exact optimal solution to the abstracted game, the exact solution has little advantage over a solution that is nearly optimal. So called e-nash equilibrium algorithms are iterative processes, which converge to the true Nash solution. Hyperborean uses an e-nash process dubbed counterfactual regret minimization to compute it's strategy. This new algorithm allowed a much larger abstraction to be solved. The 2007 version of hyperborean solved a game of size O(10^12).

Coevolution is alternative e-nash algorithm, that was initially explored by Oliehoek et al. (2005). Oliehoek et al. (2006) solved several poker variants using a coevolutionary process. The solutions presented were for very small game trees (6-8 card decks, and 1-2 rounds), but they suggested the intriguing possibility of applying coevolution to large-scale poker games. After significant work on INOT had been completed the author became aware of research by Dudziak (2006) who used coevolution to solve abstracted full scale Texas Hold'em.

3 FICTITIOUS PLAY

INOT and Fell Omen used the coevolutionary algorithm fictitious play (Brown, 1951) to determine their final strategies. Fictitious play is an iterative algorithm that proceeds as follows:

1. Both players start with a strategy. Let n=1.
2. Each player calculates the best response to the other's current strategy
3. Each player updates their strategy such that the new strategy is the result of playing the old strategy with probability (n-1)/n and the best response strategy with probability 1/n.
4. Increment n, n=n+1. Repeat steps 2-4 until convergence is achieved.

These relatively simple set of steps have been shown to converge to a Nash equilibrium is a surprisingly large set of circumstances. Brown (1951) conjectured, and Robertson (1951) proved convergence for two person zero-sum games and convergence for partnership games was established by Monderer and Shapley (1996). Though results have been formulated in other limited areas (e.g. Khrishna 1992), convergence in general is not guaranteed.

4 CONSTRUCTING THE ABSTRACTED GAME

4.1 BUCKETING

The first abstraction that must be made in solving a full-scale game is to treat similar hands identically. In the pre-flop this can be done with no loss of information. The two card hands can be divided into 169 different buckets, one for each set of suited card rank combinations, and one for each unsuited combination. Post-flop, similar hands can be grouped together on the basis raw hand strength (HS) and potential (POT) (Papp, 1998). Hand strength is defined as the probability that the hand will beat a random hand after all cards are dealt, and hand potential is defined as the probability that the hand beats its opponent after all cards have been dealt, given that the opponent currently has the best five card hand and his hand is otherwise randomly chosen. These two dimensions of similarity are used to classify hands into groups, which will be treated identically.

INOT first divided hands into buckets according to their HS. The distribution of HS at each round was calculated, and for the flop and turn 50 buckets were created dividing along the quantiles of the distribution. The river, on the other hand, was given 75 buckets. Within each bucket of the flop was then divided into three buckets using the 60^{th} and 80^{th} percentiles of POT within the HS bucket. Each turn bucket was similarly divided into two buckets based on the 75^{th} percentile of POT.

These buckets were refined for Fell Omen 1 and 2 based on INOT's final strategy. In order to have buckets that are based empirically on the strategic structure of the game, the converged strategy of INOT, which will be discussed below, was used to find areas of the HS and POT sample space where the bucketing was too course. If two adjacent buckets are different enough that it is likely that creating additional buckets in the area would be beneficial, then it is also likely that using one bucket's strategy with the other bucket's hands would lead to detectable suboptimality. For each pair of adjacent buckets, the strategy of INOT for those buckets were swapped, best response strategies were calculated, and the expected value of the best response strategies against the swapped INOT strategy was determined. Pairs with greatly reduced expected values were then singled out as candidates for further refinement.

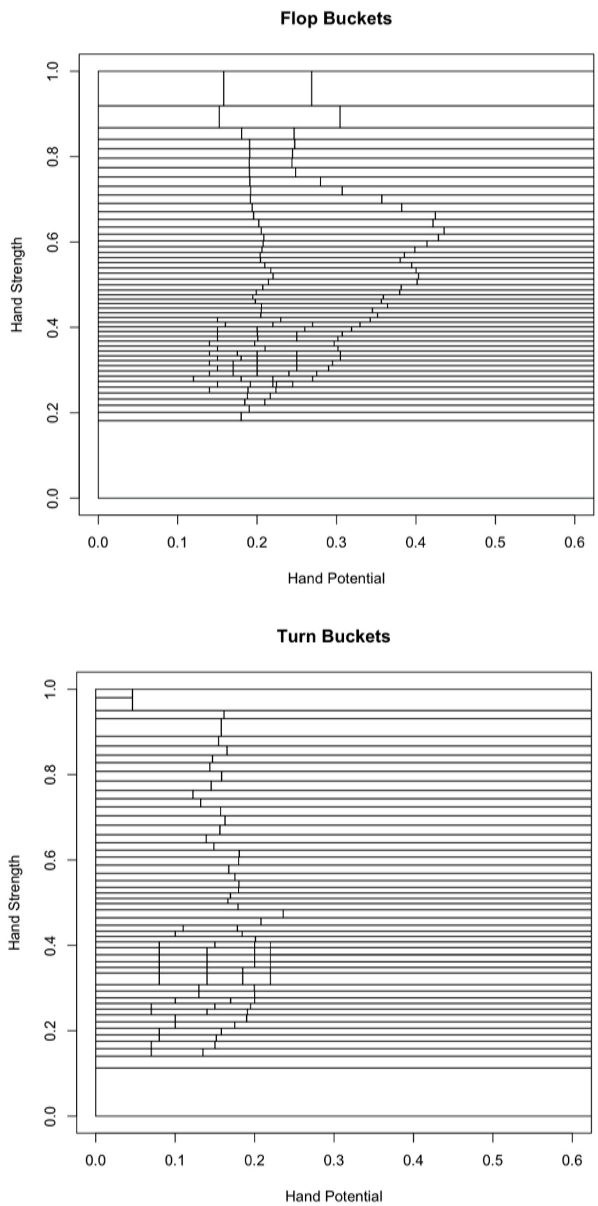

Figure 1: Bucketing Schemes for the Flop and Turn Rounds.

The number of buckets is similar to that of Dudziak (2006), and represents a significant departure from the number of buckets used by psyopti (6) and Hyperborean (up to 12). This improvement is possible because additional game tree reductions are made in the transition between betting rounds

4.2 CHANCE NODE ABSTRACTION

With the buckets defined, we now have an abstract representation of each round that represents a significant simplification over full scale Hold'em, but it is necessary to define a way to transition between betting rounds, taking into account the dealing of board cards. The transition between rounds is abstracted utilizing a matrix of probabilities. Each element of this matrix represents the probability that a player will have a hand in a certain bucket in the next round, given that they have a hand in a particular bucket in the current round. Let $b_{i,j}$ be the j^{th} bucket in the i^{th} round. 100 million seven-card hands were generated, and used to estimate three transition matrices between rounds $T_{k,j} = P(b_{i+1,k} \mid b_{i,j})$. These matrices can then be used to translate a vector of hand probabilities ($P = P(b_{i,j})$) from one round to another $P_{next} = T \cdot P$ (Dudziak, 2006), and translate a matrix of expected values ($EV_{k,l} = ev(b_{i,l}\ vs.\ b_{i,k})$) backward from one round to the previous round $EV_{previous} = T' \cdot EV \cdot T$.

In order to take into account public information, Fell Omen 2 implemented bucketed chance nodes between the flop and preflop. To determine whether a particular flop was favorable to good hand preflop hand, using the strategy from INOT, the hand probabilities given that INOT re-raised after a single preflop raise were calculated ($PRR_j = P(b_{1,j} \mid re-raise)$) and compared to the probability of being dealt that hand given INOT simply calls the big blind $PC_j = P(b_{1,j} \mid call)$. The flops were categorized into buckets based on $FlopWt = HR \cdot (PRR - PC)$, where HR is a vector with elements representing the probability that preflop hand j currently has the best five card hand (at the flop) versus a random hand, ignoring flushes. The flop was then divided into three cases: 1. All suits match, 2. Two suits match, and 3. No suits match, and cases 2 and 3 were each further subdivided into five groups based on *FlopWt*, creating a total of 11 buckets. Additional turn and river chance node buckets were investigated and determined not to provide significant improvement.

Unlike Hyperborean, our three players make decisions only based on the current hand bucket, the betting history and the flop public information buckets. Hyperborean on the other hand will play a hand of a certain hand strength and potential differently depending on whether it was a good hand that got worse, or a bad hand that got better, meaning that past private information is remembered. In the abstracted game it is very useful to forget this information, as game situations with identical public information, and identical current hand strength and potential should be treated identically.

5 CONVERGENCE

The algorithm of fictitious play was implemented in the abstracted games. Figure 2 illustrates the convergence of Fell Omen 1 to its equilibrium strategy. A strictly literal interpretation of the fictitious play algorithm was used, in that the full best response strategy was calculated at each iteration of the process. Dudziak (2006) used a pseudo-fictitious play algorithm to converge to equilibrium, only calculating a best response to one branch of the tree, holding all others constant. As a result, each best response for Fell Omen was more expensive to compute, but the strategy converged in fewer iterations.

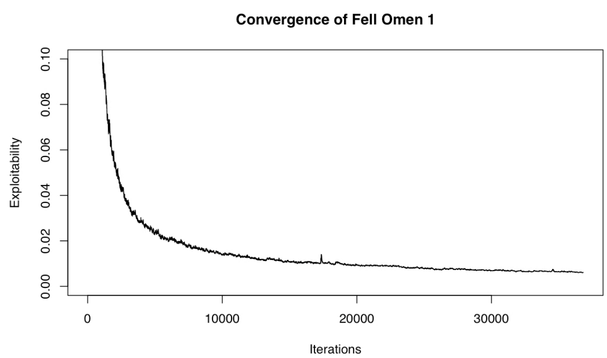

Figure 2: Convergence rate of Fell Omen 1. Exploitability in small bets per hand.

6 PERFORMANCE

INOT was entered into the AAAI 2007 Computer Poker Competition, and Fell Omen 2 was entered into the 2008 competition. They both performed well versus the other entries, coming in second both years. In 2007, INOT faced off against 15 competitors. INOT finished just behind Hyperborean (losing by .021 small bets per hand), and squeaked out a statistically insignificant victory over GS3 (winning .004 small bets per hand). The top three bots all beat the rest of the field by large margins.

INOT Win Rate versus 2007 Competitors		
Hyperborean 2007 (UoA)	GS3 (CMU)	PokeMinn 1
-0.021	0.004	0.1295
Quick	Gomel 2	Dumbo 1
0.0991	0.085	0.1415
Dumbo 2	Sequel 1	Sequel 2
0.1186	0.1307	0.14
PokeMinn 2	UNCC	Gomel 1
0.1421	0.4716	0.0875
LeRenard	Monash	Milano
0.1297	0.4078	0.3976

Table 1: AAAI 2007 Competition Results in small bets per hand (60,000 hands except 600,000 hands vs. UoA and CMU)

Fell Omen 2 was pitted against 9 opponents in 2008, and placed second in a three way tie with an aggressive version of Hyperborean (Hyperborean-On) and GGValuta created by Mihai Ciucu. The loss to Hyperborean-On and the win against GGValuta were not significant, indicating that these two competitors were extremely evenly matched vs. Fell Omen 2.

Fell Omen 2 Win Rate versus 2008 Competitors		
Hyperborean 2008-Eq (UoA)	Hyperborean 2008-On (UoA)	GGValuta
-0.024	-0.004	0.003
GS4	PokeMinn 2	PokeMinn 2
0.02	0.153	0.154
GUS	Dr. Sahbak	Gomel 1
0.467	0.532	0.0875

Table 2: AAAI 2008 Competition Results in small bets per hand (60,000 hands)

7 DISCUSSION

The performance of heads-up limit programs has improved greatly over the past years, and INOT and Fell Omen 2 have shown themselves to be competitive with the top players in the world. Fell Omen 2's loss to Hyperborean raises an interesting question. How can a program that only makes distinctions between 10-12 hand types outperform a program that has hand distinctions in the hundreds?

One possibility is that by keeping track of the past history of the current hand (i.e. it's hand strength in previous rounds), Hyperborean is capturing an additional dimension of hand type. In addition to hand strength, and potential, Papp (1997) investigated negative potential, which he defined as the probability, given that we currently have a better five card hand, the opponents hand improves, and they will have the best hand at showdown. This is a measure of how fragile the hand is, and how susceptible it is to hands with higher potential. It is possible that by retaining past private hand information, Hyperborean implicitly takes it's negative potential into account. It may be useful to introduce negative potential, and see if this increases the strength of the AI.

That said, it is unlikely that there is a great deal of room for improvement in two-player limit Hold'em, as evidenced by the fact that all of the top AIs are very evenly matched, and a version of Hyperborean beat two professional poker players in the 2008 Man Machine Challenge.

INOT and Fell Omen 1-2 have been released under open source GPL license, and are available online or from the author upon request.

References

Billings, D., Burch, N., Davidson, A., Holte, R., Schaeffer, J., Schauenberg, T., and Szafron, D. (2003). Approximating game-theoretic optimal strategies for full-scale poker. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.

Brown, G.W. (1951). Iterative Solutions of Games by Fictitious Play. Activity Analysis of Production and Allocation.

Dudziak, A. (2006) Using Fictitious Play to Find Pseudo-optimal Solutions for Full-scale Poker. International Conference on Artificial Intelligence pp 374-380.

Johanson M. (2007). Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player, M.Sc. thesis.

Koller, D., and Pfeffer, A. (1997). Representations and solutions for game-theoretic problems. Artificial Intelligence 94(1):167–215.

Monderer D., Shapley L. (1996). Fictitious play property for games with identical interests, J. Econ. Theory 68:258-265.

Krishna V. (1991). Learning in games with strategic complementarities, Harvard Business school.

Oliehoek F., Spaan M., and Vlassis N. (2005). Best-response play in partially observable card games. Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands, pp 45–50.

Oliehoek F, Vlassis N., and De Jong E. (2005). Coevolutionary Nash in poker games. Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence, 188-193.

Papp, D. (1998). Dealing with Imperfect Information in Poker, M.Sc. thesis.

Robinson J. (1951). An iterative method of solving a game, Ann. Math. 54(2):296-301.