A Texas Hold'em poker player based on automated abstraction and real-time equilibrium computation*

Andrew Gilpin
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
gilpin@cs.cmu.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
sandholm@cs.cmu.edu

ABSTRACT

We demonstrate our game theory-based Texas Hold'em poker player. To overcome the computational difficulties stemming from Texas Hold'em's gigantic game tree, our player uses automated abstraction and real-time equilibrium approximation. Our player solves the first two rounds of the game in a large off-line computation, and solves the last two rounds in a real-time equilibrium approximation. Participants in the demonstration will be able to compete against our opponent and experience first-hand the cognitive abilities of our player. Some of the techniques used by our player, which does not directly incorporate any poker-specific expert knowledge, include such poker techniques as bluffing, slow-playing, check-raising, and semi-bluffing, all techniques normally associated with human play.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: General

General Terms

Algorithms, Economics

Keywords

Keywords: game theory, equilibrium computation, game playing

1. INTRODUCTION

In environments with multiple self-interested agents, an agent's outcome is affected by actions of the other agents. Consequently, the optimal action of one agent generally depends on the actions of others. Game theory provides a normative framework for analyzing such strategic situations. In particular, game theory provides the notion of an *equilibrium*, a strategy profile in which no agent has incentive to deviate to a different strategy. Thus, it is in an agent's interest to compute equilibria of games in order to play as well as possible.

Games can be classified as either games of *perfect information* or *imperfect information*. Chess and Go are examples of the former, and, until recently, most game playing work in AI has been on games of this type. To compute an optimal strategy in a perfect information game, an agent traverses the game tree and evaluates individual nodes. If the agent is able to traverse the entire game tree, she simply computes an optimal strategy from the bottom-up, using the principle of *backward induction*. This is the main approach behind minimax with α-β-pruning. These algorithms have limits, of course, particularly when the game tree is huge, but extremely effective game-playing agents can be developed, even when the size of the game tree prohibits complete search.

Current algorithms for solving perfect information games do not apply to games of incomplete information. The distinguishing difference is that the latter are not fully observable: when it is an agent's turn to move, she does not have access to all of the information about the world. In such games, the decision of what to do at a node cannot generally be optimally made without considering decisions at all other nodes (including ones on other paths of play).

The *sequence form* is a compact representation [7, 5, 10] of a sequential game. For two-person zero-sum games, there is a natural linear programming formulation based on the sequence form that is polynomial in the size of the game tree. Thus, reasonable-sized two-person games can be solved using this method [10, 5, 6]. However, this approach still yields enormous (unsolvable) optimization problems for many real-world games, most notably poker. In this research we apply automated abstraction techniques for finding smaller, strategically similar games for which the equilibrium computation is faster. The resulting strategies can then be used as approximate solutions to the original game. We have chosen poker as the first application of our equilibrium approximation techniques.

2. POKER

Poker is an enormously popular card game played around the world. The 2005 World Series of Poker featured more than $100 million dollars in prize money in several tournaments. Increasingly, poker players compete in online poker rooms, and television stations regularly broadcast poker tournaments.

*This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

Due to the uncertainty stemming from opponents' cards, opponents' future actions, and chance moves, poker has been identified as an important research area in AI [2]. Poker has been a popular subject in the game theory literature since the field's founding, but manual equilibrium analysis has been limited to extremely small games. Very recently, there has been considerable progress in tacking larger games. In a recent paper [4], we developed automated abstraction techniques, and applied them in computing *optimal* strategies for Rhode Island Hold'em poker [9], a smaller version of Texas Hold'em that is still over four orders of magnitude larger than previously solved poker games.

2.1 Texas Hold'em

Texas Hold'em is perhaps the most popular version of poker. It is the game that is used to determine the world champion at the annual World Series of Poker. In the demonstration we will be playing *heads-up*, in which there are just 2 players (in this case, a human player versus our player). The players alternate turns being player 1 and player 2. Player 1 is considered the *small blind*, and player 2 is the *large blind*. Before any cards are dealt, the small blind contributes one chip to the pot, and the large blind contributes two chips to the pot. Both players then receive two cards each, face down; these are known as the *hole cards*.

After receiving the hole cards, the players take part in one betting round. The small blind goes first. Each player may check or bet if no bets have been placed. If a bet has been placed, then the player may *fold* (thus forfeiting the game), *call* (adding chips to the pot equal to the last player's bet), or *raise* (calling the current bet and making an additional bet). In Texas Hold'em, the players are usually limited to four raises each per betting round. In this betting round, the bets are in increments of two chips.

After the betting round, three community cards are dealt face up. These cards are called the *flop*. Another betting round take places at this point, with bets equal to two chips.

Another community card is dealt face up. This is called the *turn* card. Another betting round takes place at this point, with bets equal to four chips.

A final community card is dealt face up. This is called the *river* card. Another betting round takes place at this point, with bets equal to four chips.

If neither player folds, then the *showdown* takes place. Using the seven available cards (the two hole cards and five community cards), the players form their best 5-card poker hands. The player who has the best 5-card poker hand takes the pot. In the event of a draw, the pot is split evenly.

3. TECHNICAL OVERVIEW

The main contribution of our work is the application of automated abstraction techniques to a real-world game. Previous work has been limited to much smaller games. In this section we give a brief overview of our development of a Texas Hold'em poker player. A detailed description of our player is available in a separate paper [3]. There are two types of abstraction employed in our approach: state-space abstraction and round-based abstraction.

In our previous work [4] we developed techniques for *automatically* reducing the size of a game tree (a form of state-space abstraction) in order to make equilibrium-finding algorithms practical. We apply our algorithm, *GameShrink*, to the various game trees we encounter in the computation of strategies.

In addition to state-space abstraction, we also employ round-based abstraction. In our approach, we first solve for an approximate equilibrium for a truncated game involving only the first two rounds. We do this by solving a large linear program in an off-line computation. After the turn card appears, our player computes updated card probabilities based on observed behavior, and then computes an equilibrium approximation for the third and fourth rounds in *real-time*. The abstractions we employ in computing this equilibrium approximation are *dynamically* determined based on the information (*i.e.* community cards) revealed so far in the game. This allows our computation to focus on the specific portion of the game tree relevant to the current hand.

Round-based abstraction has been used in previous poker work [9, 1]. The primary difference with our approach is the fact that strategies are computed dynamically, using observed information to achieve a closer approximation. Furthermore, the sizes of the individual models are larger. For example, optimal strategies for pre-flop Texas Hold'em have been computed [8]. This approach requires modelling 169 distinct hands. Our model not only considers 169 hands in the first round, but also 2465 hands in the second round. Solving this model requires 18.8 GB of RAM and takes 7.1 days. In addition, our abstractions are automatically computed, rather than manually designed by an expert.

Some features of our computed strategies include poker techniques such as bluffing, slow-playing, check-raising, and semi-bluffing, all techniques normally associated with human play. In this demonstration, participants will compete with our opponent and will experience these strategies first-hand.

4. REFERENCES

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

[2] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.

[3] A. Gilpin and T. Sandholm. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. Mimeo, 2006.

[4] A. Gilpin and T. Sandholm. Finding equilibria in large sequential games of imperfect information. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, Ann Arbor, MI, 2006.

[5] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.

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

[7] I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3:678–681, 1962.

[8] A. Selby. Optimal heads-up preflop poker, 1999. http://www.archduke.demon.co.uk/simplex/.

[9] J. Shi and M. Littman. Abstraction methods for game theoretic poker. In *Computers and Games*, pages 333–345. Springer-Verlag, 2001.

[10] B. von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.