

Optimal Rhode Island Hold'em Poker*

Andrew Gilpin and Tuomas Sandholm

Computer Science Department

Carnegie Mellon University

Pittsburgh, PA 15213

{gilpin, sandholm}@cs.cmu.edu

Abstract

Rhode Island Hold'em is a poker card game that has been proposed as a testbed for AI research. This game, with a tree size larger than 3.1 billion nodes, features many characteristics present in full-scale poker (e.g., Texas Hold'em). Our research advances in equilibrium computation have enabled us to solve for the optimal (equilibrium) strategies for this game. Some features of the equilibrium include poker techniques such as bluffing, slow-playing, check-raising, and semi-bluffing. In this demonstration, participants will compete with our optimal opponent and will experience these strategies firsthand.

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 formally defines games and strategies, and provides the notion of an *equilibrium*, which is a collection of strategies (one for each player) in which no player has incentive to deviate to a different strategy (i.e., all players are playing best responses to each others' strategies). Thus, one component desired for a strong game-playing agent is an effective method for computing (or approximating) equilibria of games. The most commonly used solution concept for two-player games is *Nash equilibrium*, which is guaranteed to exist in any finite game where players are able to play randomized strategies (Nash 1950).

Games can be classified as either games of *complete information* or *incomplete 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 complete 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 and alpha-beta search. 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 complete 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 (Romanovskii 1962; Koller, Megiddo, & von Stengel 1994; von Stengel 1996) 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 (von Stengel 1996; Koller, Megiddo, & von Stengel 1996; Koller & Pfeffer 1997). However this approach still yields enormous (unsolvable) optimization problems for many real-world games, most notably poker. In this research we introduce automated abstraction techniques for finding smaller, strategically equivalent games for which the equilibrium computation is faster. We have chosen poker as the first application of our equilibrium finding techniques.

Poker

Poker is an enormously popular card game played around the world. The 2005 World Series of Poker is expected to have nearly \$50 million dollars in prize money in several tournaments. Increasingly, poker players compete in online poker rooms, and television stations regularly broadcast poker tournaments.

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 (Billings *et al.* 2002). 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. Even with the use of computers, the largest poker games that have been solved have only about 140,000 nodes (Koller & Pfeffer 1997). Large-scale approximations have been devel-

*This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.
Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

oped (Billings *et al.* 2003), but those methods do not provide any guarantees about the performance of the computed strategies. Furthermore, the approximations were designed manually by a human expert. Our approach does not require any domain knowledge.

Rhode Island Hold'em

Rhode Island Hold'em is a specific instance of poker that was invented as a testbed for AI research (Shi & Littman 2001). It was designed so that it was similar in style to Texas Hold'em, yet not so large that devising reasonably intelligent strategies would be impossible. Rhode Island Hold'em has a game tree exceeding 3.1 billion nodes, and until now finding an exact solution seemed unlikely.

Rhode Island Hold'em (in this version) is played by 2 players. Each player pays an *ante* of 5 chips which is added to the *pot*. Both players initially receive a single card, face down; these are known as the *hole cards*. After receiving the hole cards, the players take part in one betting round. 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 Rhode Island Hold'em, the players are limited to 3 raises each per betting round. In this betting round, the bets are 10 chips. After the betting round, a community card is dealt face up. This is called the *flop*. Another betting round takes place at this point, with bets equal to 20 chips. Another community card is dealt face up. This is called the *turn* card. A final betting round takes place at this point, with bets equal to 20 chips. If neither player folds, then the *showdown* takes place. Both players turn over their cards. The player who has the best 3-card poker hand takes the pot. (Hands in 3-card poker games are ranked slightly differently than 5-card poker hands. The main differences are that the order of flushes and straights are reversed, and a three of a kind is better than straights or flushes.) In the event of a draw, the pot is split evenly.

Technical contribution

This demonstration embodies the first use of our algorithm, *GameShrink*, for the automatic detection of *extensive game isomorphisms* and the application of *restricted game isomorphic abstraction transformations* (Gilpin & Sandholm 2005). Essentially, our algorithm takes as input an imperfect information game tree and outputs a strategically equivalent game that is much smaller. We can prove that a Nash equilibrium in the smaller, abstracted game is strategically equivalent to a Nash equilibrium in the original game in the sense that, given a Nash equilibrium in the abstracted game, it is simple to compute a Nash equilibrium in the original game. Thus, by shrinking the game tree, we can carry out the equilibrium computations on a smaller input, and then construct an optimal equilibria in the original game in a straightforward manner.

For even larger games, we have developed approximation technique for automatically computing abstractions that may not necessarily result in a Nash equilibrium in the smaller game. This approach does allow us to find approx-

imate equilibria for larger games. For solving Rhode Island Hold'em, we did not need to resort to such approximations.

Applying the sequence form representation to Rhode Island Hold'em yields an LP with 91,224,226 rows, and the same number of columns. This is much too large for current linear programming algorithms to handle. We used *GameShrink* to reduce this, and it yielded an LP with 1,237,238 rows and columns—with 50,428,638 non-zero coefficients in the LP. We then applied iterated elimination of dominated strategies, which further reduced this to 1,190,443 rows and 1,181,084 columns. (Applying iterated elimination of dominated strategies without *GameShrink* yielded 89,471,986 rows and 89,121,538 columns, which still would have been prohibitively large to solve.) *GameShrink* required less than one second to perform the shrinking (*i.e.*, to compute all of the restricted game isomorphic abstraction transformations). Using a 1.65GHz IBM eServer p5 570 with 64 gigabytes of RAM (we only needed 25 gigabytes), we solved the LP (and thus solved for a Nash equilibrium of the game) in 7 days and 13 hours using the barrier method of ILOG CPLEX.

While others have worked on computer programs for playing Rhode Island Hold'em (Shi & Littman 2001), no optimal strategy has been found. This is the largest poker game solved to date by over four orders of magnitude.

References

- Billings, D.; Davidson, A.; Schaeffer, J.; and Szafron, D. 2002. The challenge of poker. *Artificial Intelligence* 134(1-2):201–240.
- 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. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*.
- Gilpin, A., and Sandholm, T. 2005. Finding equilibria in large extensive form games of imperfect information. Mimeo.
- Koller, D., and Pfeffer, A. 1997. Representations and solutions for game-theoretic problems. *Artificial Intelligence* 94(1):167–215.
- Koller, D.; Megiddo, N.; and von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, 750–759.
- Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior* 14(2):247–259.
- Nash, J. 1950. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences* 36:48–49.
- Romanovskii, I. 1962. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics* 3:678–681.
- Shi, J., and Littman, M. 2001. Abstraction methods for game theoretic poker. In *Computers and Games*, 333–345. Springer-Verlag.
- von Stengel, B. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14(2):220–246.