Baby Tartanian8: Winning Agent from the 2016 Annual Computer Poker Competition

Noam Brown
Computer Science Department
Carnegie Mellon University
noamb@cs.cmu.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
sandholm@cs.cmu.edu

Abstract

Imperfect-information games, where players have private information, pose a unique challenge in artificial intelligence. In recent years, Heads-Up No-Limit Texas Hold'em poker, a popular version of poker, has emerged as the primary benchmark for evaluating game-solving algorithms for imperfect-information games. We demonstrate a winning agent from the 2016 Annual Computer Poker Competition, *Baby Tartanian8*.

1 Introduction

Imperfect-information games are used to model a variety of strategic interactions, including negotiations, auctions, security settings (both physical and digital), and recreational games such as poker. Algorithms used in perfect-information games rely on knowing the current state of the game, and then searching through the remaining reachable states with some evaluation function. Such algorithms are not applicable to imperfect-information games, because a player usually does not know the exact state of the game due to unobserved hidden information. Totally different algorithms are required to address the challenges of imperfect information.

Heads-Up (i.e., two-player) No-Limit Texas Hold'em poker has emerged as the primary benchmark for evaluating game-solving algorithms in imperfect-information games. We demonstrate one such agent. In the most recent Annual Computer Poker Competition (ACPC), in which there were 11 participants, our agent finished in 1st for the Total Bankroll Heads-Up No-Limit Texas Hold'em competition (in which agents are evaluated according to their aggregate performance against all opponents), and 3rd for the Instant Runoff Heads-Up No-Limit Texas Hold'em competition (in which the agent with the lowest bankroll performance against the remaining agents is eliminated until only one agent remains).

2 Abstraction

The version of No-Limit Texas Hold'em used in the ACPC contains 10^{165} nodes in the game tree, which is far too large for even a single traversal. The standard approach (for a review, see [Sandholm, 2010]) to constructing a strategy for such large games is to first create an *abstraction* of the game which is a manageable size but still preserves as much of the strategic characteristics of the original game as possible. The abstraction is then solved using an equilibrium-finding algorithm, and its solution mapped back to the original game.

To facilitate distributed equilibrium finding, our abstraction algorithm decomposes the game tree into disjoint parts after the early stage of the game. For *Baby Tartanian8*, we defined this "early stage" as the preflop in poker, and separated the remaining game tree into disjoint sets by conditioning on the flop. During equilibrium finding on a distributed architecture, the early stage of the game is assigned to one head node, while each remaining disjoint part is assigned to a different child node. This ensures that each machine can access memory locally and run independently, other than one message to and from the head node on each iteration. This abstraction approach was used for the top three agents in the ACPC this year, and was also used for the top agent of the previous ACPC no-limit competition [Brown *et al.*, 2015].

Our agent uses an asymmetric action abstraction, in which more actions are allowed for the opponent than for ourselves [Bard *et al.*, 2014]. This allows us to leverage domain knowledge to eliminate suboptimal actions for ourselves, while still being able to respond intelligently in case the opponent chooses suboptimal actions. Actions were selected by examining the equilibrium strategies of smaller agents and choosing the actions that were most commonly used.

3 Equilibrium Finding

For equilibrium finding, we used a distributed variant of the *Monte Carlo Counterfactual Regret Minimization algorithm (MCCFR)* based on the algorithm used by the previous ACPC no-limit winning agent *Tartanian7* [Brown *et al.*, 2015]; for earlier studies on MCCFR algorithms, see [Lanctot *et al.*, 2009; Zinkevich *et al.*, 2007]. MCCFR is an iterative algorithm which minimizes regret independently in each information set. If both players play according to MCCFR, then their average strategies provably converge to a Nash equilibrium.

Our agent also employs a novel sampling algorithm based on *Regret-Based Pruning (RBP)* [Brown and Sandholm, 2015]. RBP allows an agent to avoid exploring actions in the game tree on every iteration if those actions have performed poorly in the past, while still guaranteeing convergence to a Nash equilibrium within the same number of iterations. Thus, while the number of iterations needed to arrive within a certain ϵ of a Nash equilibrium does not change, each iteration

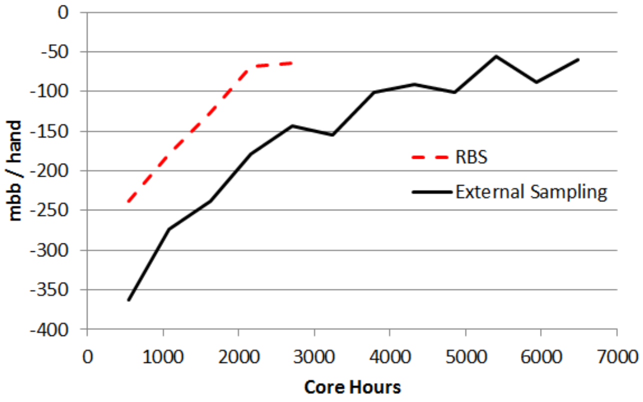

Figure 1: Performance of RBS compared to external-sampling MCCFR in a smaller-scale preliminary experiment. External sampling is the most popular form of MCCFR. Both algorithms were used to train a strategy based on identical abstractions using 64 cores. Performance in milli-big blinds per hand (mbb / hand) is shown against *Tartanian7*, the winner of the 2014 ACPC no-limit hold'em competition.

is performed far more quickly. The number of iterations for which an action may be skipped depends on how negative the regret is for that action—the action must be explored again at the earliest iteration on which its regret could turn positive. In practice, RBP leads to more than an order of magnitude improvement in the speed of convergence for small games, and this improvement appears to grow with the size of the game.

Our *sampled* implementation of RBP, which we refer to as *regret-based sampling (RBS)*, has not been proven to converge to a Nash equilibrium. Nevertheless, preliminary experiments on smaller-scale hardware, as shown in Figure 1, demonstrated a substantial increase in performance in both small and large games. Due to this strong empirical performance in large-scale experiments, we used RBS in the equilibrium finding of our competition agent, despite lacking theoretical guarantees. Although RBS likely improved our early convergence rate, there was some evidence that our implementation of RBS may have led to decreased performance when closer to convergence. For this reason, we turned off RBS for the final 5 days of the equilibrium computation.

4 Agent Construction

Our equilibrium finding was run offline at the San Diego Supercomputing Center on the *Comet* supercomputer. We used 3408 cores (142 blades with 24 cores each) for about 600 hours, for a total of about 2 million core hours. Each node had 128 GB of RAM.

Since we used an asymmetric abstraction, we solved two separate abstractions (using about 1 million core hours for each abstraction) and used half of each solution for our final agent (i.e., the first mover's strategy or the second mover's strategy). Each abstraction had $1.6 \cdot 10^{14}$ nodes in its game tree, and the final strategy required 16 TB to store as doubles.

The submission size limit for the ACPC was 200 GB. To satisfy this constraint, the final strategy was purified so that the agent would take a single action with probability one [Brown *et al.*, 2015]. The purified strategy was then compressed so that each situation would use only $\lceil \ln(|A|) \rceil$ bits to represent which action should be played, where $|A|$ is the number of possible actions in a situation. To reduce the possibility of an opponent exploiting our deterministic strategy, we did not purify or compress the early part of the game (the preflop), which requires only 170 KB to store uncompressed. The size constraints resulted in our submission of a "Baby" version of Tartanian8.

5 Demonstration

Attendees at IJCAI will have the opportunity to play against our agent in two-player No-Limit Texas Hold'em. A web interface will be used to connect to the agent. Two computers will be available for attendees to use, and the live games will be shown on large monitors as well. Based on experiments against human players and the results of the ACPC, we believe our agent plays at an expert professional human level.

6 Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1320620 and IIS-1546752, the ARO under award W911NF-16-1-0061, and the San Diego Supercomputing Center.

References

[Bard *et al.*, 2014] Nolan Bard, Michael Johanson, and Michael Bowling. Asymmetric abstractions for adversarial settings. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 501–508, 2014.

[Brown and Sandholm, 2015] Noam Brown and Tuomas Sandholm. Regret-based pruning in extensive-form games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[Brown *et al.*, 2015] Noam Brown, Sam Ganzfried, and Tuomas Sandholm. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit Texas Hold'em agent. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015.

[Lanctot *et al.*, 2009] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1078–1086, 2009.

[Sandholm, 2010] Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, pages 13–32, Winter 2010. Special issue on Algorithmic Game Theory.

[Zinkevich *et al.*, 2007] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.