Heads-up Limit Hold'em Poker is Solved

Michael Bowling,[1*] Neil Burch,[1] Michael Johanson,[1] Oskari Tammelin[2]
[1]Department of Computing Science, University of Alberta,
Edmonton, Alberta, T6G2E8, Canada
[2]Unaffiliated, http://jeskola.net

[*]To whom correspondence should be addressed; E-mail: bowling@cs.ualberta.ca

Poker is a family of games that exhibit imperfect information, where players do not have full knowledge of past events. Whereas many perfect information games have been solved (e.g., Connect Four and checkers), no nontrivial imperfect information game played competitively by humans has previously been solved. Here, we announce that heads-up limit Texas hold'em is now essentially weakly solved. Furthermore, this computation formally proves the common wisdom that the dealer in the game holds a substantial advantage. This result was enabled by a new algorithm, CFR$^+$, which is capable of solving extensive-form games orders of magnitude larger than previously possible.

Games have been intertwined with the earliest developments in computation, game theory, and artificial intelligence (AI). At the very conception of computing, Babbage had detailed plans for an "automaton" capable of playing tic-tac-toe and dreamt of his Analytical Engine playing chess (*1*). Both Turing (*2*) and Shannon (*3*) — on paper and in hardware, respectively — developed programs to play chess as a means of validating early ideas in computation and AI. For more than a half century, games have continued to act as testbeds for new ideas and the resulting successes have marked important milestones in the progress of AI. Examples include the checkers-playing computer program Chinook becoming the first to win a world championship title against humans (*4*), Deep Blue defeating Kasparov in chess (*5*), and Watson defeating Jennings and Rutter on Jeopardy! (*6*). However, defeating top human players is not the same as "solving" a game — that is, computing a game-theoretically optimal solution that is incapable of losing against any opponent in a fair game. Notable milestones in the advancement of AI have been achieved through solving games such as Connect Four (*7*) and checkers (*8*).

Every nontrivial game played competitively by humans that has been solved to date is a perfect-information game (*9*). In perfect-information games, all players are informed of everything that has occurred in the game before making a decision. Chess, checkers, and backgammon are examples of perfect-information games. In imperfect-information games, players do not always have full knowledge of past events (e.g., cards dealt to other players in bridge and poker, or a seller's knowledge of the value of an item in an auction). These games are more challenging, with theory, computational algorithms, and instances of solved games lagging behind results in the perfect-information setting (*10*). And although perfect-information may be a common property of parlor games, it is far less common in real-world decision-making settings. In a conversation recounted by Bronowski, von Neumann, the founder of modern game theory, made the same observation: "Real life is not like that. Real life consists of bluffing, of little tactics of deception, of asking yourself what is the other man going to think I mean to do. And that is what games are about in my theory" (*11*).

Von Neumann's statement hints at the quintessential game of imperfect-information: the game of poker. Poker involves each player being dealt private cards, with players taking structured turns making bets on having the strongest hand (possibly bluffing), calling opponents' bets, or folding to give up the hand. Poker played an important role in early developments in the field of game theory. Borel's (*12*) and von Neumann's (*13, 14*) foundational works were motivated by developing a mathematical rationale for bluffing in poker, and small synthetic poker games (*15*) were commonplace in many early papers (*12, 14, 16, 17*). Poker is also arguably the most popular card game in the world, with more than 150 million players worldwide (*18*).

The most popular variant of poker today is Texas hold'em. When it is played with just two players (heads-up) and with fixed bet sizes and a fixed number of raises (limit), it is called heads-up limit hold'em or HULHE (*19*). HULHE was popularized by a series of high-stakes games chronicled in the book *The Professor, the Banker, and the Suicide King* (*20*). It is also the smallest variant of poker played competitively by humans. HULHE has 3.16×10^{17} possible states the game can reach, making it larger than Connect Four and smaller than checkers. However, because HULHE is an imperfect-information game, many of these states cannot be distinguished by the acting player, as they involve information about unseen past events (i.e., private cards dealt to the opponent). As a result, the game has 3.19×10^{14} decision points where a player is required to make a decision.

Although smaller than checkers, the imperfect-information nature of HULHE makes it a far more challenging game for computers to play or solve. It was 17 years after Chinook won its first game against world champion Tinsley in checkers that the computer program Polaris won the first meaningful match against professional poker players (*21*). Whereas Schaeffer et al. solved checkers in 2007 (*8*), heads-up limit Texas hold'em poker had remained unsolved. This slow progress is not for lack of effort. Poker has been a challenge problem for artificial intelligence, operations research, and psychology, with work going back more than 40 years (*22*). 17 years ago, Koller and Pfeffer (*23*) declared, "We are nowhere close to being able to solve huge games such as full-scale poker, and it is unlikely that we will ever be able to do so." The focus on HULHE as one example of "full-scale poker" began in earnest more than 10 years ago (*24*) and became the focus of dozens of research groups and hobbyists after 2006, when it became the inaugural event in the Annual Computer Poker Competition (*25*), held in conjunction with the main conference of the Association for the Advancement of Artificial Intelligence (AAAI). This paper is the culmination of this sustained research effort toward solving a "full-scale" poker game (*19*).

Allis (*26*) gave three different definitions for solving a game. A game is said to be ultra-weakly solved if, for the initial position(s), the game-theoretic value has been determined; weakly solved if, for the initial position(s), a strategy has been determined to obtain at least the game-theoretic value, for both players, under reasonable resources; and strongly solved if, for all legal positions, a strategy has been determined to obtain the game-theoretic value of the position, for both players, under reasonable resources. In an imperfect-information game, where the game-theoretic value of a position beyond the initial position is not unique, Allis's notion of "strongly solved" is not well defined. Furthermore, imperfect-information games, because of stochasticity in the players' strategies or the game itself, typically have game-theoretic values that are real-valued rather than discretely valued (such as "win," "loss," and "draw" in chess and checkers) and are achieved in expectation over many playings of the game. As a result, game-theoretic values are often approximated, and so an additional consideration in solving a game is the degree of approximation in a solution. A natural level of approximation under which a game is essentially weakly solved is if a human lifetime of play is not sufficient to establish with statistical significance that the strategy is not an exact solution.

In this paper, we announce that heads-up limit Texas hold'em poker is essentially weakly solved. Furthermore, we bound the game-theoretic value of the game, proving that the game is a winning game for the dealer.

Solving Imperfect-Information Games

大范围的

The classical representation for an imperfect-information setting is the extensive-form game. Here the word "game" refers to a formal model of interaction between self-interested agents and applies to both recreational games and serious endeavours such as auctions, negotiation, and security. See Figure 1 for a graphical depiction of a portion of a simple poker game in extensive-form. The core of an extensive-form game is a game tree specifying branches of possible events, namely player actions or chance outcomes. The branches of the tree split at game states and each is associated with one of the players (or chance) who is responsible for determining the result of that event. The leaves of the tree signify the end of the game, and have an associated utility for each player. The states associated with a player are partitioned into information sets, which are sets of states among which the acting player cannot distinguish (e.g., corresponding to states where the opponent was dealt different private cards). The branches from states within an information set are the player's available actions. A strategy for a player specifies for each information set a probability distribution over the available actions. If the game has exactly two players and the utilities at every leaf sum to zero, the game is called zero-sum.

The classical solution concept for games is a Nash equilibrium, a strategy for each player such that no player can increase his or her expected utility by unilaterally choosing a different strategy. All finite extensive-form games have at least one Nash equilibrium. In zero-sum games, all equilibria have the same expected utilities for the players, and this value is called

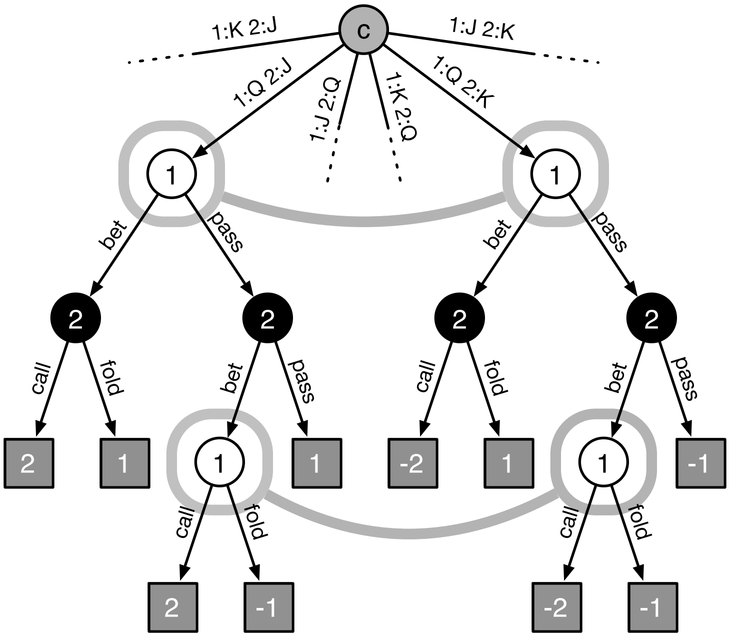


Figure 1: **Portion of the extensive-form game representation of three-card Kuhn poker (*16*).** Player 1 is dealt a queen (Q) and the opponent is given either the jack (J) or king (K). Game states are circles labeled by the player acting at each state ("c" refers to chance, which randomly chooses the initial deal). The arrows show the events the acting player can choose from, labeled with their in-game meaning. The leaves are square vertices labeled with the associated utility for player 1 (player 2's utility is the negation of player 1's). The states connected by thick gray lines are part of the same information set; that is, player 1 cannot distinguish between the states in each pair because they each represent a different unobserved card being dealt to the opponent. Player 2's states are also in information sets, containing other states not pictured in this diagram.

the game-theoretic value of the game. An ϵ-Nash equilibrium is a strategy for each player where no player can increase his or her utility by more than ϵ by choosing a different strategy. By Allis's categories, a zero-sum game is ultra-weakly solved if its game-theoretic value is computed, and weakly solved if a Nash equilibrium strategy is computed. We call a game essentially weakly solved if an ϵ-Nash equilibrium is computed for a sufficiently small ϵ to be statistically indistinguishable from zero in a human lifetime of played games. For perfect-information games, solving typically involves a (partial) traversal of the game tree. However, the same techniques cannot apply to imperfect-information settings. We briefly review the advances in solving imperfect-information games, benchmarking the algorithms by their progress in solving increasingly larger synthetic poker games as summarized in Figure 2.

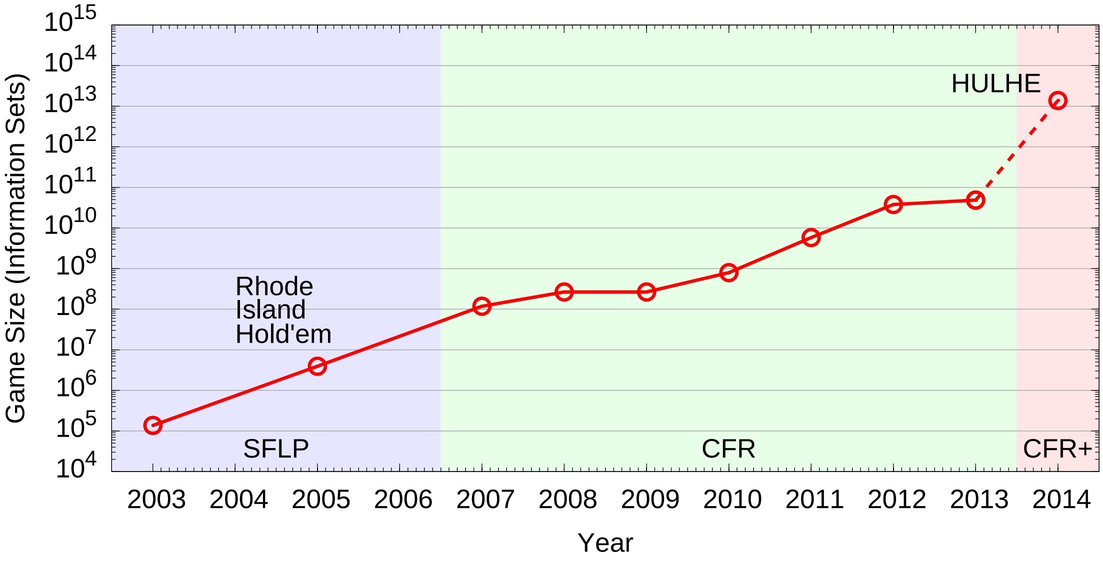


Figure 2: **Increasing sizes of imperfect-information games solved over time measured in unique information sets (i.e., after symmetries are removed).** The shaded regions refer to the technique used to achieve the result; the dashed line shows the result established in this paper.

Normal-Form Linear Programming The earliest method for solving an extensive-form game involved converting it into a normal-form game, represented as a matrix of values for every pair of possible deterministic strategies in the original extensive-form game, and then solving it with a linear program (LP). Unfortunately, the number of possible deterministic strategies is exponential in the number of information sets of the game. So, although LPs can handle normal-form games with many thousands of strategies, even just a few dozen decision points makes this method impractical. Kuhn poker, a poker game with three cards, one betting round, and a one-bet maximum having a total of 12 information sets (see Figure 1), can be solved with this approach. But even Leduc hold'em (*27*), with six cards, two betting rounds, and a two-bet maximum having a total of 288 information sets, is intractable having more than 10^{86} possible deterministic strategies.

Sequence-Form Linear Programming Romanovskii (*28*) and later Koller et al. (*29, 30*) established the modern era of solving imperfect-information games, introducing the sequence-form representation of a strategy. With this simple change of variables, they showed that the extensive-form game could be solved directly as an LP, without the need for an exponential conversion to normal-form. A sequence-form linear program (SFLP) was the first algorithm to solve imperfect-information extensive-form games with computation time that grows as a polynomial of the size of the game representation. In 2003, Billings et al. (*24*) applied this technique to poker, solving a set of simplifications of HULHE to build the first competitive poker-playing program. In 2005, Gilpin and Sandholm (*31*) used the approach along with an automated technique for finding game symmetries to solve Rhode Island hold'em (*32*), a synthetic poker game with 3.94×10^6 information sets after symmetries are removed.

Counterfactual Regret Minimization In 2006, the Annual Computer Poker Competition was started (*25*). The competition drove advancements in solving larger and larger games, with multiple techniques and refinements being proposed in the years that followed (*33, 34*). One of the techniques to emerge, and currently the most widely adopted in the competition, is counterfactual regret minimization (CFR) (*35*). CFR is an iterative method for approximating a Nash equilibrium of an extensive-form game through the process of repeated self-play between two regret-minimizing algorithms (*19, 36*). Regret is the loss in utility an algorithm suffers for not having selected the single best deterministic strategy, which can only be known in hindsight. A regret-minimizing algorithm is one that guarantees that its regret grows sub-linearly over time, and so eventually achieves the same utility as the best deterministic strategy. The key insight of CFR is that instead of storing and minimizing regret for the exponential number of deterministic strategies, CFR stores and minimizes a modified regret for each information set and subsequent action, which can be used to form an upper bound on the regret for any deterministic strategy. An approximate Nash equilibrium is retrieved by averaging each player's strategies over all of the iterations, and the approximation improves as the number of iterations increases. The memory needed for the algorithm is linear in the number of information sets, rather than quadratic, which is the case for efficient LP methods (*37*). Because solving large games is usually bounded by available memory, CFR has resulted in an increase in the size of solved games similar to that of Koller et al.'s advance. Since its introduction in 2007, CFR has been used to solve increasingly complex simplifications of HULHE, reaching as many as 3.8×10^{10} information sets in 2012 (*38*).

Solving Heads-Up Limit Hold'em

The full game of HULHE has 3.19×10^{14} information sets. Even after removing game symmetries, it has 1.38×10^{13} information sets (i.e., three orders of magnitude larger than previously solved games). There are two challenges for established CFR variants to handle games at this scale: memory and computation. During computation, CFR must store the resulting solution and the accumulated regret values for each information set. Even with single-precision (4-byte) floating-point numbers, this requires 262 TB of storage. Furthermore, past experience has shown that an increase in the number of information sets by three orders of magnitude requires at least three orders of magnitude more computation. To tackle these two challenges we use two ideas recently proposed by a co-author of this paper (*39*).

symmetrie
对称性,

To address the memory challenge we store the average strategy and accumulated regrets using compression. We use fixed-point arithmetic by first multiplying all values by a scaling factor and truncating them to integers. The resulting integers are then ordered to maximize compression efficiency, with compression ratios around 13-to-1 on the regrets and 28-to-1 on

arithmetic
算术

the strategy. Overall, we require less than 11 TB of storage to store the regrets and 6 TB to store the average strategy during the computation, which is distributed across a cluster of computation nodes. This amount is infeasible to store in main memory, and so we store the values on each node's local disk. Each node is responsible for a set of subgames; that is portions of the game tree are partitioned on the basis of publicly observed actions and cards such that each information set is associated with one subgame. The regrets and strategy for a subgame are loaded from disk, updated, and saved back to disk, using a streaming compression technique that decompresses and recompresses portions of the subgame as needed. By making the subgames large enough, the update time dominates the total time to process a subgame. With disk pre-caching, the inefficiency incurred by disk storage is approximately 5% of the total time.

To address the computation challenge we use a variant of CFR called CFR$^+$ (*19, 39*). CFR implementations typically sample only portions of the game tree to update on each iteration. They also use regret matching at each information set, which maintains regrets for each action and chooses among actions with positive regret with probability proportional to that regret. By contrast, CFR$^+$ does exhaustive iterations over the entire game tree and uses a variant of regret matching (regret matching$^+$) where regrets are constrained to be non-negative. Actions that have appeared poor (with less than zero regret for not having been played) will be chosen again immediately after proving useful (rather than waiting many iterations for the regret to become positive). Finally, unlike with CFR, we have empirically observed that the exploitability of the players' current strategies during the computation regularly approaches zero. Therefore, we can skip the step of computing and storing the average strategy, instead using the players' current strategies as the CFR$^+$ solution. We have empirically observed CFR$^+$ to require considerably less computation, even when computing the average strategy, than state-of-the-art sampling CFR (*40*), while also being highly suitable for massive parallelization.

Like CFR, CFR$^+$ is an iterative algorithm that computes successive approximations to a Nash equilibrium solution. The quality of the approximation can be measured by its exploitability: the amount less than the game value that the strategy achieves against the worst-case opponent strategy in expectation (*19*). Computing the exploitability of a strategy involves computing this worst-case value, which traditionally requires a traversal of the entire game tree. This was long thought to be intractable for games the size of HULHE. Recently it was shown that this calculation could be accelerated by exploiting the imperfect-information structure of the game and regularities in the utilities (*41*). This is the technique we use to confirm the approximation quality of our resulting strategy. The technique and implementation has been verified on small games and against independent calculations of the exploitability of simple strategies in HULHE.

A strategy can be exploitable in expectation and yet, because of chance elements in the game and randomization in the strategy, its worst-case opponent still is not guaranteed to be winning after any finite number of hands. We define a game to be essentially solved if a lifetime of play is unable to statistically differentiate it from being solved at 95% confidence. Imagine someone playing 200 games of poker an hour for 12 hours a day without missing a day for 70 years. Furthermore imagine that player using the worst-case, maximally exploitive, opponent strategy and never making a mistake. The player's total winnings, as a sum of many millions of independent outcomes, would be normally distributed. Hence, the observed winnings in this lifetime of poker would be 1.64 standard deviations or more below its expected value (i.e., the strategy's exploitability) at least 1 time out of 20. Using the standard deviation of a single game of HULHE, which has been reported to be around 5 bb/g (big-blinds per game, where the big-blind is the unit of stakes in HULHE) (*42*), we arrive at a threshold of $(1.64 * 5)/\sqrt{200 * 12 * 365 * 70} \approx 0.00105$. Therefore, an approximate solution with an exploitability less than 1mbb/g (milli-big-blinds per game) cannot be distinguished with high confidence from an exact solution, and indeed has a 1-in-20 chance of winning against its worst-case adversary even after a human lifetime of games. Hence, 1mbb/g is the threshold for declaring HULHE essentially solved.

The Solution

Our CFR$^+$ implementation was executed on a cluster of 200 computation nodes each with 24 2.1-GHz AMD cores, 32GB of RAM, and a 1-TB local disk. We divided the game into 110,565 subgames (partitioned according to preflop betting, flop cards, and flop betting). The subgames were split among 199 worker nodes, with one parent node responsible for the initial portion of the game tree. The worker nodes performed their updates in parallel, passing values back to the parent node for it to perform its update, taking 61 minutes on average to complete one iteration. The computation was then run for 1579 iterations, taking 68.5 days, and using a total of 900 core-years of computation (*43*) and 10.9 TB of disk space, including filesystem overhead from the large number of files.

Figure 3 shows the exploitability of the computed strategy with increasing computation. The strategy reaches an exploitability of 0.986 mbb/g, making HULHE essentially weakly solved. Using the separate exploitability values for each position (as the dealer and non-dealer) we get exact bounds on the game-theoretic value of the game: between 87.7 and 89.7 mbb/g for the dealer, proving the common wisdom that the dealer holds a substantial advantage in HULHE.

The final strategy, as a close approximation to a Nash equilibrium, can also answer some fundamental and long-debated questions about game-theoretically optimal play in HULHE. Figure 4 gives a glimpse of the final strategy in two early decisions of the game. Human players have disagreed about whether it may be desirable to "limp" (i.e., call as the very first action rather than raise) with certain hands. Conventional wisdom is that limping forgoes the opportunity to provoke an immediate fold by the opponent, and so raising is preferred. Our solution emphatically agrees (see the absence of blue in Figure 4a). The strategy limps just 0.06% of the time and with no hand more than 0.5%. In other situations, the strategy gives insights beyond conventional wisdom, indicating areas where humans might improve. The strategy almost never "caps" (i.e., makes the final allowed raise) in the first round as the dealer, whereas some strong human players cap the betting with a wide range of hands. Even when holding the strongest hand — a pair of aces — the strategy caps the betting less than 0.01% of

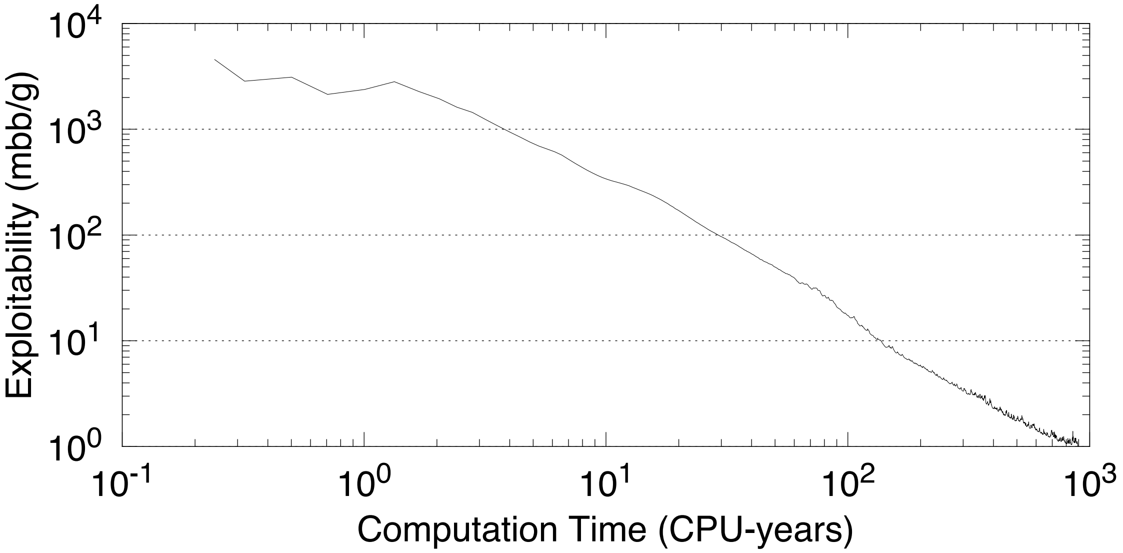


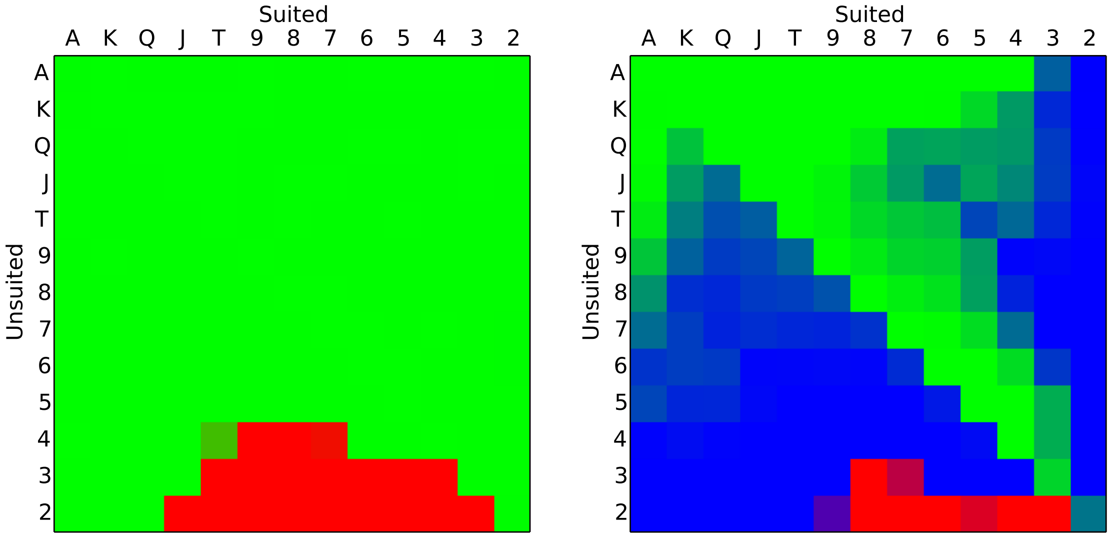
Figure 3: **Exploitability of the approximate solution with increasing computation.** The exploitability, measured in milli-big-blinds per game (mbb/g), is that of the current strategy measured after each iteration of CFR$^+$. After 1579 iterations or 900 core-years of computation, it reaches an exploitability of 0.986 mbb/g.

the time, and the hand most likely to cap is a pair of twos, with probability 0.06%. Perhaps more important, the strategy chooses to play (i.e., not fold) a broader range of hands as the non-dealer than most human players (see the relatively small amount of red in Figure 4b). It is also much more likely to re-raise when holding a low-rank pair (such as threes or fours) (*44*).

Although these observations are for only one example of game-theoretically optimal play (different Nash equilibria may play differently), they confirm as well as contradict current human beliefs about equilibria play, and illustrate that humans can learn considerably from such large-scale game-theoretic reasoning.

Conclusion

What is the ultimate importance of solving poker? The breakthroughs behind our result are general algorithmic advances that make game-theoretic reasoning in large-scale models of any sort more tractable. And, although seemingly playful, game theory has always been envisioned to have serious implications [e.g., its early impact on Cold War politics (*45*)]. More recently, there has been a surge in game-theoretic applications involving security, including systems being deployed for airport checkpoints, air marshal scheduling, and coast guard patrolling (*46*). CFR algorithms based on those described above, have been used for robust decision-making in settings where there is no apparent adversary, with potential application to medical decision



(a) first action as the dealer

(b) first action as the non-dealer after a dealer raise

Figure 4: **Action probabilities in the solution strategy for two early decisions.** (A) The action probabilities for the dealers first action of the game. (B) The action probabilities for the nondealers first action in the event that the dealer raises. Each cell represents one of the possible 169 hands (i.e., two private cards), with the upper right diagonal consisting of cards with the same suit and the lower left diagonal consisting of cards of different suits. The color of the cell represents the action taken: red for fold, blue for call, and green for raise, with mixtures of colors representing a stochastic decision.

support (*47*). With real-life decision-making settings almost always involving uncertainty and missing information, algorithmic advances such as those needed to solve poker the key to future applications. However, we also echo a response attributed to Turing in defense of his own work in games: "It would be disingenuous of us to disguise the fact that the principal motive which prompted the work was the sheer fun of the thing" (*48*).

References and Notes

1. C. Babbage, *Passages from the Life of a Philosopher* (Longman, Green, Longman, Roberts, and Green, London, 1864). Chapter 34.

2. A. Turing, *Faster Than Thought*, B. V. Bowden, ed. (Pitman, London, 1976), chap. 25.

3. C. E. Shannon, *Philosophical Magazine, Series 7* **41**, 256 (1950).

4. J. Schaeffer, R. Lake, P. Lu, M. Bryant, *AI Magazine* **17**, 21 (1996).

5. M. Campbell, A. J. Hoane Jr., F. Hsu, *Artificial Intelligence* **134**, 57 (2002).

6. D. Ferrucci, *IBM Journal of Research and Development* **56**, 1:1 (2012).

7. V. Allis, A knowledge-based approach to connect-four. the game is solved: White wins, Master's thesis, Vrije Universiteit (1988).

8. J. Schaeffer, *et al.*, *Science* **317**, 1518 (2007).

9. We use the word trivial to describe a game that can be solved without the use of a machine. The one near-exception to this claim is oshi-zumo, but it is not played competitively by humans and is a simultaneous-move game that otherwise has perfect information (*49*). Furthermore, almost all nontrivial games played by humans that have been solved to-date also have no chance elements. The one notable exception is hypergammon, a three-checker variant of backgammon invented by Sconyers in 1993, which he then strongly solved (i.e., the game-theoretic value is known for all board positions). It has seen play in human competitions (see http://www.bkgm.com/variants/HyperBackgammon.html).

10. For example, Zermelo proved the solvability of finite, two-player, zero-sum, perfect information games in 1913 (*50*), whereas von Neumann's more general minimax theorem appeared in 1928 (*13*). Minimax and alpha-beta pruning, the fundamental computational algorithm for perfect-information games, were developed in the 1950s; the first polynomial-time technique for imperfect-information games was introduced in the 1960s but was not well known until the 1990s (*29*).

11. J. Bronowski, The ascent of man, Documentary (1973). Episode 13.

12. É. Borel, J. Ville, *Applications de la théorie des probabilités aux jeux de hasard* (Gauthier-Villars, Paris, 1938).

13. J. von Neumann, *Mathematische Annalen* **100**, 295 (1928).

14. J. von Neumann, O. Morgenstern, *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, Second Edition, 1947).

15. We use the word synthetic to describe a game that was invented for the purpose of being studied or solved rather than played by humans. A synthetic game may be trivial, such as Kuhn poker (*16*), or nontrivial such as Rhode Island hold'em (*32*).

16. H. Kuhn, *Contributions to the Theory of Games I*, H. Kuhn, A. Tucker, eds., Annals of mathematics studies (Princeton University Press, 1950), pp. 97–103.

17. J. F. Nash, L. S. Shapley, *Contributions to the Theory of Games I*, H. Kuhn, A. Tucker, eds. (Princeton University Press, 1950), pp. 105–116.

18. *The Economist* **December 22**, 31 (2007).

19. See Supplementary Materials.

20. M. Craig, *The Professor, the Banker, and the Suicide King: Inside the Richest Poker Game of All Time* (Grand Central Publishing, New York City, 2006).

21. J. Rehmeyer, N. Fox, R. Rico, *Wired* **16.12**, 186 (2008).

22. D. Billings, A. Davidson, J. Schaeffer, D. Szafron, *Artificial Intelligence* **134**, 201 (2002).

23. D. Koller, A. Pfeffer, *Artificial Intelligence* **94**, 167 (1997).

24. D. Billings, *et al.*, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (2003), pp. 661–668.

25. M. Zinkevich, M. Littman, *Journal of the International Computer Games Association* **29**, 166 (2006). News item.

26. V. L. Allis, Searching for solutions in games and artificial intelligence, Ph.D. thesis, University of Limburg (1994).

27. F. Southey, *et al.*, *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (2005), pp. 550–558.

28. I. V. Romanovskii, *Soviet Mathematics* **3**, 678 (1962).

29. D. Koller, N. Megiddo, *Games and Economic Behavior* **4**, 528 (1992).

30. D. Koller, N. Megiddo, B. von Stengel, *Games and Economic Behavior* **14**, 247 (1996).

31. A. Gilpin, T. Sandholm, *Journal of the ACM* **54** (2007).

32. J. Shi, M. L. Littman, *Revised Papers from the Second International Conference on Computers and Games* (2000), pp. 333–345.

33. T. Sandholm, *AI Magazine* **31**, 13 (2010).

34. J. Rubin, I. Watson, *Artificial Intelligence* **175**, 958 (2011).

35. Another notable algorithm to emerge from the Annual Computer Poker Competition is an application of Nesterov's excessive gap technique (*51*) to solving extensive-form games (*52*). The technique has some desirable properties, including better asymptotic time complexity than what is known for CFR. However, it has not seen widespread use among competition participants because of its lack of flexibility in incorporating sampling schemes and its inability to be used with powerful (but unsound) abstractions that make use of imperfect recall. Recently, Waugh and Bagnell (*53*) have shown that CFR and the excessive

gap technique are more alike than different, which suggests that the individual advantages of each approach may be attainable in the other.

36. M. Zinkevich, M. Johanson, M. Bowling, C. Piccione, *Advances in Neural Information Processing Systems 20* (2008), pp. 905–912.

37. N. Karmarkar, *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing* (ACM, 1984), pp. 302–311.

38. E. Jackson, *Proceedings of the 2012 Computer Poker Symposium* (2012). Jackson reports a higher number of information sets, which counts terminal information sets rather than only those where a player is to act.

39. O. Tammelin, *CoRR* **abs/1407.5042** (2014).

40. M. Johanson, N. Bard, M. Lanctot, R. Gibson, M. Bowling, *Proceedings of the Eleventh International Conference on Autonomous Agents and Multi-Agent Systems* (2012), pp. 837–846.

41. M. Johanson, K. Waugh, M. Bowling, M. Zinkevich, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence* (2011), pp. 258–265.

42. M. Bowling, M. Johanson, N. Burch, D. Szafron, *Proceedings of the Twenty-Fifth International Conference on Machine Learning* (2008), pp. 72–79.

43. The total time and number of core-years is larger than was strictly necessary as it includes computation of an average strategy that was later measured to be more exploitable than the current strategy and so was discarded. The total space noted, on the other hand, is without storing the average strategy.

44. These insights were the result of discussions with Bryce Paradis, previously a professional poker player who specialized in HULHE.

45. O. Morgenstern, *New York Times Magazine* pp. 21–22 (1961).

46. M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned* (Cambridge University Press, 2011).

47. K. Chen, M. Bowling, *Advances in Neural Information Processing Systems 25* (2012), pp. 2078–2086.

48. P. Mirowski, *Toward a History of Game Theory*, E. R. Weintraub, ed. (Duke University Press, Durham, North Carolina, 1992), pp. 113–147. Mirowski cites Turing as author of the paragraph containing this remark. The paragraph appeared in (*2*), in a chapter with Turing listed as one of three contributors. Which parts of the chapter are the work of which

contributor, particularly the introductory material containing this quote, is not made explicit.

49. M. Buro, *Advances in Computer Games* **135**, 361 (2004).

50. E. Zermelo, *Proceedings of the Fifth International Congress of Mathematicians* (Cambridge University Press, Cambridge, 1913), pp. 501–504.

51. Y. Nesterov, *SIAM Journal on Optimization* **16**, 233 (2005).

52. A. Gilpin, S. Hoda, J. Peña, T. Sandholm, *Proceedings of the Third International Workshop On Internet And Network Economics* (2007), pp. 57–69.

53. K. Waugh, J. A. Bagnell, *AAAI Workshop on Computer Poker and Imperfect Information* (2015).

Acknowledgments

The author order is alphabetical reflecting equal contribution by the authors. The idea of CFR$^+$ and compressing the regrets and strategy originated with O. T. (*39*). This research was supported by Natural Sciences and Engineering Research Council of Canada and Alberta Innovates Technology Futures through the Alberta Innovates Centre for Machine Learning, and was only made possible by the computing resources of Compute Canada and Calcul Québec. We thank all of the current and past members of the University of Alberta Computer Poker Research Group, where the idea to solve heads-up limit Texas hold'em was first discussed; J. Schaeffer, R. Holte, D. Szafron, and A. Brown for comments on early drafts of this article; and B. Paradis for insights into the conventional wisdom of top human poker players.

Supplementary Materials

Source code used to compute the solution strategy

Supplementary Text

Fig. S1

References (54–62)