Supplementary Materials for

Heads-up Limit Hold’em Poker is Solved

The Game of Heads-Up Limit Hold’em

Heads-up limit Texas hold’em is a two-player poker game. It is a repeated game, in which the two players typically play a match of individual games, usually called *hands*, while alternating who is the *dealer*. In each of the individual games, one player will win some number of *chips* from the other player, and the goal is to win as many chips as possible over the course of the match.

Each individual game begins with both players placing a number of chips in the *pot*: the player in the dealer position puts in the *small blind*, and the other player puts in the *big blind*, which is twice the small blind amount. The game then progresses through four rounds: the *preflop*, *flop*, *turn*, and *river*. Each round consists of cards being dealt followed by player actions in the form of wagers as to who will hold the strongest hand at the end of the game. In the preflop, each player is given two private cards, unobserved by their opponent. In the later rounds, cards are dealt face-up in the center of the table, called *community cards*. A total of five community cards are revealed over the four rounds: three on the flop, one on the turn, and one on the river.

After the cards for the round are dealt players alternate taking actions from among exactly three options: fold, call, or raise. A player *folds* by declining to match the last opponent wager, thus forfeiting to the opponent all chips in the pot and ending the game with no player revealing their private cards. A player *calls* by adding chips into the pot to match the last opponent wager, which causes the next round to begin. A player *raises* by adding chips into the pot to match the last wager followed by adding additional chips to make a wager of their own. At the beginning of a round when there is no opponent wager yet to match, the raise action is called *bet*, and the call action is called *check*, which only ends the round if both players check. The size of all wagers are fixed: in the preflop and flop rounds the size is called the *small-bet* and is equal to the big blind; on the turn and river rounds the size is called the *big-bet* and is twice the big blind. The non-dealer takes the first action in each round except in the pre-flop where the dealer must decide whether to fold, call, or raise the non-dealer’s big blind bet. A maximum of four bets or raises are allowed in each round, with the initial blinds counting as a single bet in the pre-flop. Once the fourth bet or raise is made, the responding player may only call or fold.

If the river round ends with no player previously folding to end the game, the outcome is determined by a *showdown*. Each player reveals their two private cards and the player that can form the strongest five-card poker hand (see "List of poker hands" on Wikipedia; accessed July 21, 2014) wins all the chips in the pot. To form their hand each player may use any cards from their two private cards and the five community cards At the end of the game, whether ended by fold or showdown, the players will swap who is the dealer and begin the next game.

Since the game can be played for different stakes, such as a big blind being worth $0.01 or $1 or $1000, players commonly measure their performance over a match as their average number of big blinds won per game. Researchers have standardized on the unit *milli-big-blinds per game*, or mbb/g, where one milli-big-blind is $\frac{1}{1000}$ of one big blind. A player that always folds will lose 750 mbb/g (by losing 1000 mbb as the big blind and 500 as the small blind). A human rule-of-thumb is that a professional should aim to win at least 50 mbb/g from their opponents. Milli-big-blinds per game is also used as a unit of exploitability, when it is computed as the expected loss per game against a worst-case opponent.

Poker Glossary

big-bet The size of all bets or raises in the 3^{rd} or 4^{th} round. A big-bet is twice the size of a small-bet.

big blind Initial wager made by the non-dealer before any cards are dealt. The big blind is twice the size of the small blind.

bet Starting a new wager in a round, putting more chips into the pot.

call Putting enough chips into the pot to match the current wager; ends the round.

cap Make the fourth bet or raise in a round. In the first round, the blinds are considered the first bet.

check Declining to start a new wager in a round.

chip Marker representing value used for wagers, comes in various denominations including the small blind, the big blind, and possibly others.

community cards Public cards, dealt face up, visible to all players. Used in combination with hole cards to create a hand.

dealer The player who puts the small blind into the pot. Acts first on round 1, and second on the later rounds. Traditionally, they would distribute community cards and hole cards from the deck.

flop The 2^{nd} round; can refer to either the 3 revealed community cards, or the betting round after these cards are revealed.

fold Give up on the current hand, forfeiting all wagers placed in the pot. Ends a player's participation in the game.

game A single playing of poker. Ends after all but one player folds or a showdown is reached.

hand Many different meanings: the combination of the best 5 cards from the community cards and hole cards, just the hole cards themselves, or a single game of poker (for clarity, we avoid this final meaning).

hole cards Private cards, dealt face down, visible only to one player. Used in combination with community cards to create a hand.

limp Start the game by calling the big blind rather than raising.

milli-big-blinds per game (mbb/g) Average winning rate over a number of hands, measured in thousandths of big blinds.

pot The collected chips from all wagers.

preflop The 1^{st} round; can refer to either the hole cards, or the betting round after these cards are distributed.

raise Increasing the size of a wager in a round, putting more chips into the pot than is required to call the current bet.

river The 4^{th} and final round; can refer to either the 1 revealed community card, or the betting round after this card is revealed.

showdown After the river, players who have not folded show their hole cards to determine the player with the best hand. The player with the best hand takes all of the chips in the pot.

small-bet The size of all bets or raises in the 1^{st} or 2^{nd} round. A small-bet is equal in size to the big blind.

small blind Initial wager made by the dealer before any cards are dealt. The small blind is half the size of the big blind.

stakes The size of the blinds in a match of poker games.

turn The 3^{rd} round; can refer to either the 1 revealed community card, or the betting round after this card is revealed.

The History of Solving HULHE

Poker has been used as a research testbed for artificial intelligence and game theory for over 60 years. Over the last decade, the game of HULHE has emerged as a standard variant for evaluating and comparing artificial intelligence algorithms for playing imperfect information games. Our solving of HULHE builds on eleven years of research focused on the game.

The research community began to focus on HULHE as a challenge problem in 2003, when the Computer Poker Research Group at the University of Alberta created the first strategy intended to approximate a Nash equilibrium for the game (*24*). Using the Sequence Form Linear Programming technique of Romanovskii (*28*) and Koller et al. (*29, 30*), HULHE was far too large to be solved directly. Instead, Billings et al. employed the abstraction principle first explored by Shi and Littman (*32*). This approach has now been formalized as the Abstraction-Solving-Translation approach and underlies most of the subsequent game theoretic programs applied in the poker domain. First, the full-scale poker game is modelled by a smaller, tractably-sized abstract game. Second, this abstract game is solved to find a close approximation to an abstract game Nash equilibrium. Third, as needed, this abstract strategy is translated to choose actions in the real game. In order to produce an abstract game that can be tractably solved, the abstraction process cannot perfectly model the real game, and the resulting strategy will be exploitable in the real game. It is not currently known how exploitable this first game theoretic strategy for HULHE by Billings et al. is; however, as it was soundly defeated in games against later strategies, it was demonstrably not a very close equilibrium approximation for the real game.

In 2006, research groups at the University of Alberta and Carnegie Mellon University coordinated to form the Annual Computer Poker Competition (ACPC), an event held in conjunction with the Association for the Advancement of Artificial Intelligence (AAAI) conference. HULHE was the founding event in 2006, and has been played in every year of the ACPC. Since 2006, the HULHE events have been entered by 59 research groups and a total of 120 competing programs. Until 2013, the competition ran two HULHE events: the Bankroll event in which players aim to maximize their total winnings against the field of opponents, and the Instant Runoff event in which losing players are iteratively eliminated. This second competition is well aligned with the goal of solving the game, as a Nash equilibrium would not lose on expectation against any opponent, and so would at worst tie for first place.

Since the inaugural event of the ACPC, researchers have developed a succession of game solving algorithms that allow for ever-larger games to be solved (*36,40,52,54–58,62*). The ability to solve a larger game means that a finer-grained abstraction can be used, which more closely models the real game of HULHE. The resulting larger abstract game strategies have typically outperformed earlier strategies based on necessarily smaller, coarser abstractions. However, there is no theoretical guarantee that finer-grained abstractions will result in a better approximation of a Nash equilibrium, and, in fact, counterexamples exist (*59*).

In 2011, Johanson et al. (*41*) developed an accelerated best response computation that, for the first time, made it feasible to measure a HULHE strategy's approximation quality by efficiently computing its exploitability. This allowed researchers to revisit the previous five years of efforts and measure the progress made over time towards approximating a Nash equilibrium. In Figure S1, we show the exploitability of the known least exploitable poker programs in each year from 2006 to 2013, culminating in the CFR$^+$ solution developed in this work. Four of these strategies are noteworthy. The strategy shown in 2006 was for a technique that preceded CFR, called Range of Skill (*62*). In 2007 and 2008, the University of Alberta hosted two Man-vs-Machine Poker Championships in which their agent, Polaris, competed against human poker professionals. In 2007, Polaris narrowly lost its match against Phil Laak and Ali Eslami. In 2008, an improved version of Polaris narrowly won against a team of professionals that included Matt Hawrilenko, a HULHE specialist. Although Polaris was competitive with human professionals, we now know that it was still quite far from optimal play, being exploitable for 275.9 mbb/g and 235.3 mbb/g respectively.

With the development of the accelerated best response computation, several ACPC competitors cooperated with the University of Alberta in 2011 to measure the exploitability of their 2010 agents. While the University of Alberta's agent Hyperborean placed third in the 2010 competition, it was the least exploitable of the evaluated programs at 135.4 mbb/g. The first-place program Rockhopper, developed by David Lin, was the most exploitable of the top three programs at 300.0 mbb/g, and second-place GGValuta, developed at the University of Bucharest by Mihai Ciucu et al., was exploitable for 237.3 mbb/g. This experiment served both to benchmark the progress of the poker research community, and to note the potential inconsistency between winning competitions and minimizing exploitability prior to achieving the level of essentially solved.

Finally, with the introduction of CFR$^+$ in 2014 by Tammelin (*39*) and our joint effort to use it to solve HULHE, we have concluded this 11-year effort to solve the game. Since CFR$^+$ allows us to scale to the size of HULHE and solve the game directly, it does not require any of the abstraction and translation techniques that have historically been necessary for earlier algorithms to imperfectly approximate a Nash equilibrium. As the ACPC typically only achieves statistical significance of 5-10 mbb/g, the strategy described in this paper would in the worst-case tie for first in any Instant Runoff event.

The CFR$^+$ Algorithm

The CFR$^+$ algorithm is a variant of counterfactual regret minimization (CFR). Here we formally describe CFR before describing the modifications for CFR$^+$. We begin by giving mathematical notation to the extensive form game formulation from the main text.

Notation

An extensive form game is a tuple $\langle N, H, A, Z, P, u_{i\in N}, \mathcal{I}_{i\in N}\rangle$. N is the set of players. The game tree is represented as a set of histories, H. Each individual history $h \in H$ is a game state

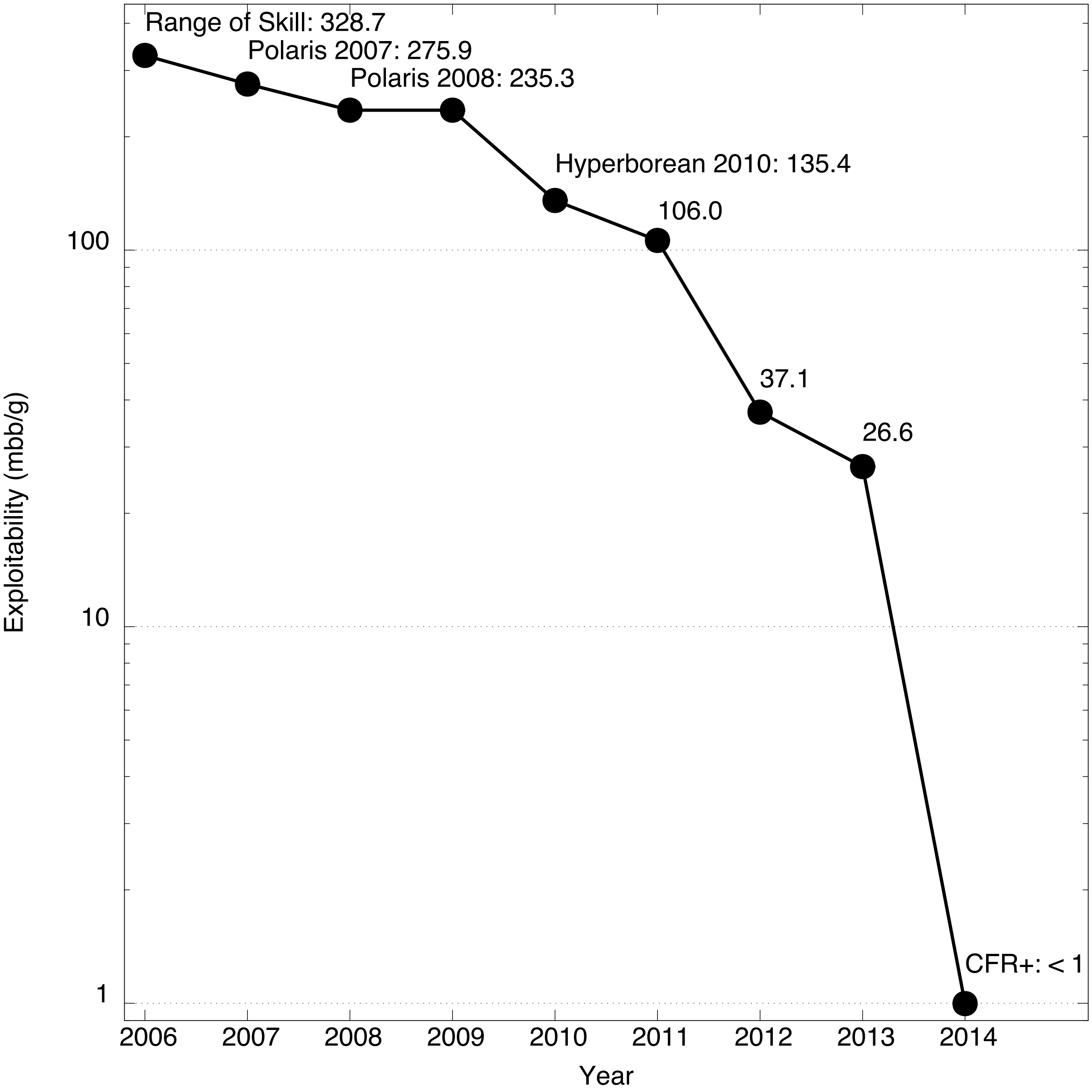

Figure S1: The decreasing exploitability of computer poker strategies for HULHE. Notable datapoints are described in the text.

and defined as a sequence of actions from the set A. H forms a tree so that if $ha \in H$ then $h \in H$, and is called the parent of ha. We define $h \sqsubset h'$ to mean that the history h is a proper prefix of the history h', or equivalently h is an ancestor of h' in the game tree. $Z \subset H$ is the set of terminal histories (i.e., leaves of the tree), where $z \in Z$ is not a parent of any history. $P(h) \in N \cup \{c\}$ is a function assigning a player to every non-terminal history $h \in H \setminus Z$, specifying who selects the next action; where c is the chance player and $\sigma_c(a|h)$ is the fixed and known probability that chance selects action a at history h. The functions $u_i(z) \in \mathbb{R}$ specify the utility to player $i \in N$ for every terminal history $z \in Z$, and is required to be bounded, so $|u_i(z) - u_i(z')| \leq \Delta$ for all $i \in N$ and $z, z' \in Z$. Finally, $\mathcal{I}_i$ is a partition of the set of histories $H_i \equiv \{h \in H | P(h) = i\}$. A block of the partition $I \in I_i$ is an information set, where two histories are in the same information set if and only if player i cannot distinguish between the two histories. For any history $h \in H_i$, define $I^h \in \mathcal{I}_i$ to be the information set containing history h, and for any history $h \in H_c$, define $I^h \equiv h$.

A game is two-player zero-sum if $N = \{1, 2\}$ and $u_2(z) = -u_1(z)$ for all $z \in Z$. A game has perfect recall if and only if for all histories h, h' in the same information set I for player i, if we let $pa \sqsubset h$ and $p'a' \sqsubset h'$ be the longest prefixes of h and h' (respectively) such that $P(p) = P(p') = i$, it must be that $a = a'$ and $\{p, p'\} \subseteq I'$ for some information set $I' \in \mathcal{I}_i$. In other words, if two histories cannot be distinguished by a player, then the last time the same player acted, they must be similarly indistinguishable; or more simply, a player cannot forget what they previously knew. CFR is guaranteed to converge to a Nash equilibrium in any two-player, zero-sum game with perfect recall (*36*), and has been observed experimentally to produce strong strategies in some non-two-player (*60*), non-zero-sum (*41*), and imperfect recall games (*61*).

A (behavioral) strategy for player i is denoted σ_i, where $\sigma_i(a|I)$ gives the probability of player i choosing action a at information set $I \in \mathcal{I}_i$ under strategy σ_i. Let Σ_i denote the set of such strategies, and $\sigma \in \times_{i \in N} \Sigma_i$ denote a strategy profile, i.e., a strategy for each player. Let (σ_i, σ'_{-i}) be a strategy profile where player i follows their strategy in σ and the other players follow their strategy in σ'. Finally, for any information set $I \in \mathcal{I}_i$, define $\sigma : I \to a$ to be the strategy profile where player i selects the action a with probability 1 in information set $I \in \mathcal{I}_i$, but in all other histories (and all other players) choose actions according to σ.

Define $\pi^\sigma(z) \equiv \prod_{ha \sqsubset z} \sigma_{P(h)}(a|I^h)$ to be the probability of reaching terminal history z when players choose actions according to σ. We can further break down this product into one player's contribution, $\pi_i^\sigma(z) \equiv \prod_{ha \sqsubset z : P(h) = i} \sigma_i(a|I^h)$, and all other players' (including chance's) contribution, $\pi_{-i}^\sigma(z) \equiv \prod_{ha \sqsubset z : P(h) \neq i} \sigma_{P(h)}(a|I^h)$. Now, we can write $u_i(\sigma) = \sum_z \pi^\sigma(z) u_i(z)$ to be the expected utility to player i when players follow profile σ. Furthermore, we can describe an ϵ-Nash equilibrium as a strategy profile σ^* where,

$$u_i(\sigma^*) \geq u_i(\sigma'_i, \sigma^*_{-i}) - \epsilon \qquad \forall \sigma'_i \in \Sigma_i \quad \forall i \in N. \tag{1}$$

Counterfactual Regret Minimization (CFR)

Let σ be any strategy profile, the counterfactual value of player i taking action a at information set I is defined as,

$$v_i^{\sigma}(I,a) \equiv \sum_{h \in I} \sum_{\substack{z \in Z: \\ h \sqsubset z}} u_i(z) \pi_{-i}^{\sigma}(z) \pi_i^{\sigma:I \to a}(h,z). \tag{2}$$

In other words, it is the expected utility to player i of reaching information set I and taking action a, under the counterfactual assumption that player i takes actions to do so, but otherwise player i and all other players follow the strategy profile σ. Let $\{\sigma^1, \ldots, \sigma^T\}$ be a sequence of strategy profiles. The counterfactual regret of player i for action a at information set I is then defined as,

$$R_i^T(I,a) \equiv \sum_{t=1}^{T} v_i^{\sigma^t}(I,a) - \sum_{t=1}^{T} \sum_{a' \in A} v_i^{\sigma^t}(I,a') \sigma_i^t(a'|I), \tag{3}$$

i.e., the amount of additional counterfactual value that player i could have attained in expectation if action a was chosen deterministically rather than the actual sequence of strategies chosen.

Counterfactual regret minimization (CFR) is an iterative self-play algorithm where the players at every information set follow a regret-minimizing algorithm to choose actions. The common choice for such an algorithm is regret-matching. In regret-matching the player i defines their strategy at time t, σ_i^t, as a function of the sequence of strategy profiles up to time $t-1$,

$$\sigma_i^t(a|I) = \begin{cases} \frac{\left(R_i^{t-1}(I,a)\right)^+}{\sum_{a' \in A}\left(R_i^{t-1}(I,a')\right)^+} & \text{if } \sum_{a' \in A}\left(R_i^{t-1}(I,a')\right)^+ > 0 \\ \frac{1}{|A|} & \text{otherwise} \end{cases}, \tag{4}$$

where $(x)^+ \equiv \max(0,x)$. By following regret-matching, the following is guaranteed,

$$R_i^T(I,a) \leq \Delta\sqrt{|A|T}, \tag{5}$$

i.e., the counterfactual regret at every information set grows sublinearly with the number of iterations, T.

Given a sequence of strategy profiles we are particularly interested in a player's overall regret given the sequence. Overall regret is defined as,

$$R_i^T \equiv \max_{\sigma_i \in \Sigma_i} \left(\sum_{t=1}^{T} u_i(\sigma_i, \sigma_{-i}^t) \right) - \sum_{t=1}^{T} u_i(\sigma) \tag{6}$$

i.e., the amount of additional utility that player i could have attained in expectation if they had chosen the best fixed strategy in hindsight rather than the actual sequence of strategies chosen. For a game with perfect recall, this quantity can be bounded by CFR's per-information-set counterfactual regrets,

$$R_i^T \leq \sum_{I \in \mathcal{I}_i} \max_{a \in A} R_i^T(I,a) \leq |\mathcal{I}_i| \Delta \sqrt{|A|T}. \tag{7}$$

CFR is therefore a regret-minimizing algorithm (i.e., achieves sublinear regret) for extensive form games.

The regret-minimizing property can be leveraged to make CFR into a game solving algorithm. It is well-known that in a two-player zero-sum game with perfect recall if $R_i^T \leq \epsilon$ for all $i \in \{1, 2\}$, then the average strategy profile $\bar{\sigma}$ is a 2ϵ-Nash equilibrium. Each players' average strategy, $\bar{\sigma}_i$ has to be constructed so that it achieves the same expected utility as the uniform mixture of strategies $(\sigma_i^1, \ldots, \sigma_i^T)$. This can be done by combining the strategies in the mixture to select an action at an information set in proportion to that strategy's probability of playing to reach that information set, so,

$$\bar{\sigma}_i(a|I) \equiv \frac{\sum_{t=1}^T \left(\sum_{h \in I} \pi_i^{\sigma^t}(h)\right) \sigma^t(a|I)}{\sum_{t=1}^T \left(\sum_{h \in I} \pi_i^{\sigma^t}(h)\right)}, \tag{8}$$

where the term $\sum_{h \in I} \pi_i^{\sigma^t}(h)$ is player i's contribution to the probability of reaching a history in information set I, and is thus the weighting term on σ_i^t. By running CFR for enough iterations, ϵ can be driven arbitrarily small.

This presentation described vanilla CFR, which exhaustively computes exact counterfactual values and regrets for every pair of information sets and actions on each iteration. The literature contains a number of variants that employ sampling to compute noisy estimates of the counterfactual values and regrets. They trade off faster iterations for a larger number of iterations, often resulting in considerably reduced overall time. CFR$^+$ is built around exhaustive iterations, and so we do not discuss sampling any further.

CFR$^+$

Now that we have described CFR, CFR$^+$ can be described quite succinctly: replace regret-matching with regret-matching$^+$. Regret-matching$^+$ replaces Equation 4 for choosing the strategy. The algorithm instead maintains alternative counterfactual regret values $Q^t(I, a)$ updated as follows,

$$\begin{aligned} &Q^0(I, a) = 0 \\ &Q^t(I, a) = \left(Q^{t-1}(I, a) + R^t(I, a) - R^{t-1}(I, a)\right)^+ \end{aligned} \tag{9}$$

The value $Q^t(I, a)$ is updated with the same change to the actual counterfactual regrets, except if this update gives a negative result then it is thresholded at zero. This prevents these alternative counterfactual regret values from ever becoming negative. Furthermore, any future positive regret changes will immediately add to these alternative values rather than cancelling out accumulated negative regret. The CFR$^+$ algorithm then selects actions according to regret-matching using these values,

$$\sigma_i^{t+1}(a|I) = \begin{cases} \frac{Q_i^t(I,a)}{\sum_{a' \in A} Q_i^t(I,a')} & \text{if } \sum_{a' \in A} Q_i^t(I, a') > 0 \\ \frac{1}{|A|} & \text{otherwise} \end{cases}, \tag{10}$$

where no zero thresholding need occur since $Q^t(I,a)$ is always non-negative.

When using CFR, the exploitability of the current strategy σ^t typically does not approach zero. It is the average strategy $\bar{\sigma}^t$, computed and stored separately from the current strategy, that is the solution returned by CFR. When using CFR$^+$, it is still sound to use the average strategy $\bar{\sigma}^t$, but we also have observed empirically that the exploitability of the current strategy σ^t regularly approaches zero. As a result, a practitioner may skip the computation and storage of the average strategy and use the current strategy as the computed solution, just as we did with our result. This is a very attractive quality of CFR$^+$ offering a significant savings in computation and storage. It is not known under what conditions the exploitability of the current strategy converges to zero or if there might be some sound weighted average strategy that would improve on the solution quality of the current strategy.

Exploitability and ϵ-Nash Equilibria

In two-player zero-sum games, it is easy to relate the Nash equilibrium approximation quality of a strategy profile to the exploitability of the strategies in that profile. Let v be the game value, i.e., the utility to the first player under any Nash equilibrium, and let $\hat{v}$ be the utility to the first player if both players play the approximate solution strategy under consideration. Let ϵ_i be the amount of utility that the ith player can gain by unilaterally choosing a different strategy in the approximate solution. For $\epsilon_i \leq \epsilon$, we say the approximate solution is an ϵ-Nash equilibrium. Finally, let x_i be the exploitability of the ith player's strategy in the approximate solution. Since asymmetric games typically involve players alternating positions in repeated playings, the exploitability of the approximate solution is taken to be the average: $x = \frac{1}{2}(x_1 + x_2)$. By the definition of exploitability, the player's utility under the opponent's best unilateral deviation cannot be less than the strategy's exploitability. Mathematically,

$$\hat{v} - \epsilon_2 \geq v - x_1 \tag{11}$$

$$-\hat{v} - \epsilon_1 \geq -v - x_2 \tag{12}$$

Adding both sides of these inequalities and using the definition of ϵ-Nash,

$$\epsilon = \max\{\epsilon_1, \epsilon_2\} \leq \epsilon_1 + \epsilon_2 \leq x_1 + x_2 = 2x \tag{13}$$

Thus, the Nash approximation ϵ is at most twice the strategy's exploitability x.

Source Code

The source code used to achieve the result described in this work is available as part of the supplementary online materials. The most recent release of the code can be retrieved from `http://poker.cs.ualberta.ca/software.html`.

References

54. M. Lanctot, K. Waugh, M. Zinkevich, M. Bowling, *Advances in Neural Information Processing Systems 22* (2009), pp. 1078–1086.

55. S. Hoda, A. Gilpin, J. Peña, T. Sandholm, *Mathematics of Operations Research* **35**, 494 (2010).

56. A. Gilpin, J. Peña, T. Sandholm, *Mathematical Programming* **133**, 279 (2012).

57. M. Johanson, N. Bard, N. Burch, M. Bowling, *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence* (2012), pp. 1371–1379.

58. N. Burch, M. Johanson, M. Bowling, *Proceedings of the Twenty-Eighth Conference on Artificial Intelligence* (2014), pp. 602–608.

59. K. Waugh, D. Schnizlein, M. Bowling, D. Szafron, *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems* (2009), pp. 781–788.

60. R. Gibson, Regret minimization in games and the development of champion multiplayer computer poker-playing agents, Ph.D. thesis, University of Alberta (2013).

61. K. Waugh, *et al.*, *Proceedings of the Eighth Symposium on Abstraction, Reformulation and Approximation* (2009), pp. 175–182.

62. M. Zinkevich, M. Bowling, N. Burch, *Proceedings of the Twenty-Second Conference on Artificial Intelligence* (2007), pp. 788–793.