Safe and Nested Endgame Solving for Imperfect-Information Games

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

Unlike perfect-information games, imperfect-information games cannot be decomposed into subgames that are solved independently. Thus more computationally intensive equilibrium-finding techniques are used, and *abstraction*—in which a smaller version of the game is generated and solved—is essential. *Endgame solving* is the process of computing a (presumably) better strategy for just an endgame than what can be computationally afforded for the full game. Endgame solving has many benefits, such as being able to 1) solve the endgame in a finer information abstraction than what is computationally feasible for the full game, and 2) incorporate into the endgame actions that an opponent took that were not included in the action abstraction used to solve the full game. We introduce an endgame solving technique that outperforms prior methods both in theory and practice. We also show how to adapt it, and past endgame-solving techniques, to respond to opponent actions that are outside the original action abstraction; this significantly outperforms the state-of-the-art approach, action translation. Finally, we show that endgame solving can be repeated as the game progresses down the tree, leading to significantly lower exploitability. All of the techniques are evaluated in terms of exploitability; to our knowledge, this is the first time that exploitability of endgame-solving techniques has been measured in large imperfect-information games.

Introduction

Imperfect-information games model strategic settings that have hidden information. They have a myriad of applications such as negotiation, shopping agents, cybersecurity, physical security, and so on. In such games, the typical goal is to find a Nash equilibrium, which is a profile of strategies—one for each player—such that no player can improve her outcome by unilaterally deviating to a different strategy.

Endgame solving is a standard technique in perfect-information games such as chess and checkers (Bellman 1965). In fact, in checkers it is so powerful that it was used to solve the entire game (Schaeffer et al. 2007).

In imperfect-information games, endgame solving is drastically more challenging. In perfect-information games it is possible to solve just a part of the game in isolation, but this is not generally possible in imperfect-information games. For example, in chess, determining the optimal response to the Queen's Gambit requires no knowledge of the optimal response to the Sicilian Defense. To see that such a decomposition is not possible in imperfect-information games, consider the game of Coin Toss shown in Figure 1. In that game, a coin is flipped and lands either Heads or Tails with equal probability, but only Player 1 sees the outcome. Player 1 can then choose between actions Left and Right, with Left leading to some unknown subtree. If Player 1 chooses Right, then Player 2 has the opportunity to guess how the coin landed. If Player 2 guesses correctly, Player 1 receives a reward of -1 and Player 2 receives a reward of 1 (the figure shows rewards for Player 1; Player 2 receives the negation of Player 1's reward). Clearly Player 2's optimal strategy depends on the probabilities that Player 1 chooses Right with Heads and Tails. But the probability that Player 1 chooses Right with Heads depends on what Player 1 could alternatively receive by choosing Left instead. So it is not possible to determine what Player 2's optimal strategy is in the Right subtree without knowledge of the Left subtree.

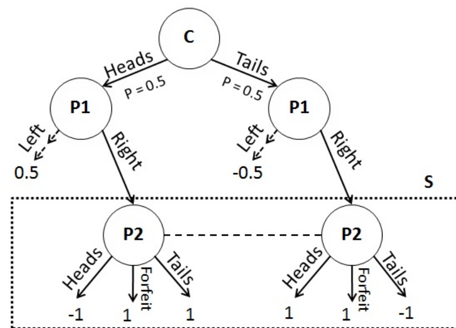

Figure 1: The example game of Coin Toss. "C" represents a chance node. S is a Player 2 (P_2) information set. The dotted line between the two P_2 nodes means P_2 cannot distinguish between the two states.

Thus imperfect-information games cannot be solved via decomposition as perfect-information games can. Instead, the entire game is typically solved as a whole. This is a problem for large games, such as No-Limit Texas Hold'em—a common benchmark problem in imperfect-information game solving—which has 10^{165} nodes (Johanson 2013). The standard approach to computing strategies in such large games is to first generate an *abstraction* of the game, which is a smaller version of the game that retains as much as possible the strategic characteristics of the original game (Sandholm 2010). This abstract game is solved (exactly or approximately) and its solution is mapped back to the original game. In extremely large games, a small abstraction typically cannot capture all the strategic complexity of the game, and therefore results in a solution that is not a Nash equilibrium when mapped back to the original game. For this reason, it seems natural to attempt to improve the strategy when a sequence farther down the game tree is reached and the remaining subtree of reachable states is small enough to be represented without any abstraction (or in a finer abstraction), even though—as explained previously—this may not lead to a Nash equilibrium. While it may not be possible to arrive at an equilibrium by analyzing subtrees independently, it may be possible to *improve* the strategies in those subtrees when the original (base) strategy is suboptimal, as is typically the case when abstraction is applied.

We first review prior forms of endgame solving for imperfect-information games. Then we propose a new form of endgame solving that retains the theoretical guarantees of the best prior methods while performing better in practice. Finally, we introduce a method for endgame solving to be nested as players descend the game tree, leading to substantially better performance.

Notation and Background for Imperfect-Information Games

In an imperfect-information extensive-form game there is a finite set of players, $\mathcal{P}$. H is the set of all possible histories (nodes) in the game tree, represented as a sequence of actions, and includes the empty history. $A(h)$ is the actions available in a history and $P(h) \in \mathcal{P} \cup c$ is the player who acts at that history, where c denotes chance. Chance plays an action $a \in A(h)$ with a fixed probability $\sigma_c(h,a)$ that is known to all players. The history h' reached after an action is taken in h is a child of h, represented by $h \cdot a = h'$, while h is the parent of h'. If there exists a sequence of actions from h to h', then h is an ancestor of h' (and h' is a descendant of h). $Z \subseteq H$ are terminal histories for which no actions are available. For each player $i \in \mathcal{P}$, there is a payoff function $u_i : Z \rightarrow \Re$. If $P = \{1,2\}$ and $u_1 = -u_2$, the game is two-player zero-sum.

Imperfect information is represented by *information sets* (infosets) for each player $i \in \mathcal{P}$ by a partition $\mathcal{I}_i$ of $h \in H : P(h) = i$. For any infoset $I \in \mathcal{I}_i$, all histories $h, h' \in I$ are indistinguishable to player i, so $A(h) = A(h')$. $I(h)$ is the infoset I where $h \in I$. $P(I)$ is the player i such that $I \in \mathcal{I}_i$. $A(I)$ is the set of actions such that for all $h \in I$, $A(I) = A(h)$. $|A_i| = \max_{I \in \mathcal{I}_i} |A(I)|$ and $|A| = \max_i |A_i|$.

A strategy $\sigma_i(I)$ is a probability vector over $A(I)$ for player i in infoset I. The probability of a particular action a is denoted by $\sigma_i(I,a)$. Since all histories in an infoset belonging to player i are indistinguishable, the strategies in each of them must be identical. That is, for all $h \in I$, $\sigma_i(h) = \sigma_i(I)$ and $\sigma_i(h,a) = \sigma_i(I,a)$. A full-game strategy $\sigma_i \in \Sigma_i$ defines a strategy for each infoset belonging to Player i. A strategy profile σ is a tuple of strategies, one for each player. $u_i(\sigma_i, \sigma_{-i})$ is the expected payoff for player i if all players play according to the strategy profile $\langle \sigma_i, \sigma_{-i} \rangle$.

$\pi^\sigma(h) = \Pi_{h' \cdot a \sqsubseteq h} \sigma_{P(h)}(h,a)$ is the joint probability of reaching h if all players play according to σ. $\pi^\sigma_i(h)$ is the contribution of player i to this probability (that is, the probability of reaching h if all players other than i, and chance, always chose actions leading to h). $\pi^\sigma_{-i}(h)$ is the contribution of all players other than i, and chance. $\pi^\sigma(h,h')$ is the probability of reaching h' given that h has been reached, and 0 if $h \not\sqsubset h'$. In a *perfect-recall* game, $\forall h, h' \in I \in \mathcal{I}_i$, $\pi_i(h) = \pi_i(h')$. In this paper we focus specifically on two-player zero-sum perfect-recall games. Therefore, for $i = P(I)$ we define $\pi_i(I) = \pi_i(h)$ for $h \in I$. Moreover, $I' \sqsubset I$ if for some $h' \in I'$ and some $h \in I$, $h' \sqsubset h$. Similarly, $I' \cdot a \sqsubset I$ if $h' \cdot a \sqsubset h$. We also define $\pi^\sigma(I, I')$ as the probability of reaching I' from I according to the strategy σ.

For convenience, we define an endgame. If a history is in an endgame, then any other history with which it shares an infoset must also be in the endgame. Moreover, any descendent of the history must be in the endgame. Formally, an *endgame* is a set of histories $S \subseteq H$ such that for all $h \in S$, if $h \sqsubset h'$, then $h' \in S$, and for all $h \in S$, if $h' \in I(h)$ for some $I \in \mathcal{I}_{P(h)}$ then $h' \in S$. The *head* of an endgame S_r is the union of infosets that have actions leading directly into S, but are not in S. Formally, S_r is a set of histories such that for all $h \in S_r$, $h \notin S$ and either $\exists a \in A(h)$ such that $h \rightarrow a \in S$, or $h \in I$ and for some history $h' \in I$, $h' \in S_r$.

A *Nash equilibrium* (Nash 1950) is a strategy profile σ^* such that $\forall i,\ u_i(\sigma^*_i, \sigma^*_{-i}) = \max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma^*_{-i})$. An *$\epsilon$-Nash equilibrium* is a strategy profile σ^* such that $\forall i,\ u_i(\sigma^*_i, \sigma^*_{-i}) + \epsilon \geq \max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma^*_{-i})$. In two-player zero-sum games, every Nash equilibrium results in the same expected value for a player. A *best response* $BR_i(\sigma_{-i})$ is a strategy for player i such that $u_i(BR_i(\sigma_{-i}), \sigma_{-i}) = \max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma_{-i})$. The *exploitability* $exp(\sigma_{-i})$ of a strategy σ_{-i} is defined as $u_i(BR_i(\sigma_{-i}), \sigma_{-i}) - u_i(\sigma^*)$, where σ^* is a Nash equilibrium.

A *counterfactual best response* (Moravcik et al. 2016) $CBR_i(\sigma_{-i})$ is similar to a best response, but additionally maximizes counterfactual value at every infoset. Specifically, a counterfactual best response is a strategy σ_i that is a best response with the additional condition that if $\sigma_i(I,a) > 0$ then $v^\sigma_i(I,a) = \max_{a'} v^\sigma(I,a')$.

We further define *counterfactual best response value* $CBV^{\sigma_{-i}}(I)$ as the value player i expects to achieve by playing according to $CBR_i(\sigma_i)$ when in infoset I. Formally $CBV^{\sigma_{-i}}(I,a) = \sum_{h \in I} \left(\pi^{\sigma_{-i}}_{-i}(h) \sum_{z \in Z} \left(\pi^{\langle CBR_i(\sigma_{-i}), \sigma_{-i} \rangle}(h \cdot a, z) u_i(z) \right) \right)$ and $CBV^{\sigma_{-i}}(I) = \max_{a \in A(I)} CBV^{\sigma_{-i}}(I,a)$.

Prior Approaches to Endgame Solving in Imperfect-Information Games

In this section we review prior techniques for endgame solving in imperfect-information games. Our new algorithm then builds on some of the ideas and notation.

Throughout this section, we will refer to the Coin Toss game shown in Figure 1. We will focus on the Right endgame. If P_1 chooses Left, the game continues to a much larger endgame, but its structure is not relevant here.

We assume that a *base* strategy profile σ has already been computed for this game in which P_1 chooses Right $\frac{3}{4}$ of the time with Heads and $\frac{1}{2}$ of the time with Tails, and P_2 chooses Heads $\frac{1}{2}$ of the time, Tails $\frac{1}{4}$ of the time, and Forfeit $\frac{1}{4}$ of the time after P_1 chooses Right. The details of the base strategy in the Left endgame are not relevant in this section, but we assume that if P_1 played optimally then she would receive an expected payoff of 0.5 for choosing Left if the coin is Heads, and -0.5 for choosing Left if the coin is Tails. We will attempt to improve P_2's strategy in the endgame that follows P_1 choosing Right. We refer to this endgame as S.

Unsafe Endgame Solving

We first review the most intuitive form of endgame solving, which we refer to as *unsafe endgame solving* (Billings et al. 2003; Gilpin and Sandholm 2006; 2007; Ganzfried and Sandholm 2015). This form of endgame solving assumes that both players will play according to their base strategies outside of the endgame. In other words, all nodes outside the endgame are fixed and can be treated as chance nodes with probabilities determined by the base strategy. Thus, the different roots of the endgame are reached with probabilities determined from the base strategies using Bayes' rule. A strategy is then computed for the endgame—independently from the rest of the game. Applying unsafe endgame solving to Coin Toss (after P_1 chooses Right) would mean solving the game shown in Figure 2.

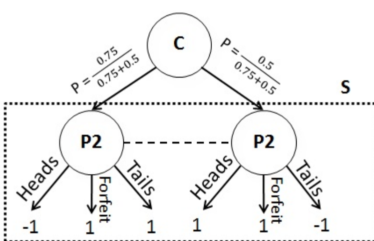

Figure 2: The game solved by Unsafe endgame solving to determine a P_2 strategy in the Right endgame of Coin Toss.

Specifically, we define R as the set of earliest-reachable histories in S. That is, $h \in R$ if $h \in S$ and $h' \notin S$ for any $h' \sqsubset h$. We then calculate $\pi^\sigma(h)$ for each $h \in R$. A new game is constructed consisting only of an initial chance node and S. The initial chance node reaches $h \in R$ with probability $\frac{\pi^\sigma(h)}{\sum_{h' \in R} \pi^\sigma(h')}$. This new game is solved and its strategy is then used whenever S is encountered.

Unsafe endgame solving lacks theoretical solution quality guarantees and there are many situations where it performs extremely poorly. Indeed, if it were applied to the base strategy of Coin Toss, it would produce a strategy in which P_2 always chooses Heads—which P_1 could exploit severely by only choosing Right with Tails. Despite the lack of theoretical guarantees and potentially bad performance, unsafe endgame solving is simple and can *sometimes* produce low-exploitability strategies in large games, as we show later.

We now move to discussing *safe* endgame solving techniques, that is, ones that ensure that the exploitability of the strategy is no higher than that of the base strategy.

Re-Solve Refinement

In *Re-solve refinement* (Burch, Johanson, and Bowling 2014), a safe strategy is computed for P_2 in the endgame by constructing an *auxiliary game*, as shown in Figure 3, and computing an equilibrium strategy σ^S for it. The auxiliary game consists of a starting chance node that connects to each history h in S_r in proportion to the probability that player P_1 could reach h if P_1 tried to do so (that is, in proportion to $\pi^\sigma_{-1}(h)$). Let a_S be the action available in h such that $h \cdot a_S \in S$. At this point, P_1 has two possible actions. Action a'_S, the auxiliary-game equivalent of a_S, leads into S, while action a'_T leads to a terminal payoff that awards the counterfactual best response value from the base strategy $CBV^{\sigma_{-1}}(I(h), a_S)$. In the base strategy of Coin Toss, the counterfactual best response value of P_1 choosing Right is 0 if the coin is Heads and $\frac{1}{2}$ if the coin is Tails. Therefore, a'_T leads to a terminal payoff of 0 for Heads and $\frac{1}{2}$ for Tails. After the equilibrium strategy σ^S is computed in the auxiliary game, σ^S_2 is copied back to S in the original game (that is, P_2 plays according to σ^S_2 rather than σ_2 when in S). In this way, the strategy for P_2 in S is pressured to be similar to that in the original strategy; if P_2 were to choose a strategy that did better than the base strategy against Heads but worse against Tails, then P_1 would simply choose a'_T with Heads and a'_S with Tails.

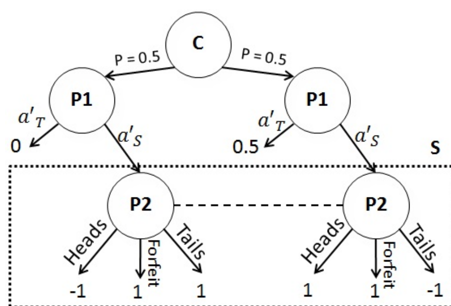

Figure 3: The auxiliary game used by Re-solve refinement to determine a P_2 strategy in the Right endgame of Coin Toss.

Re-solve refinement is safe and useful for compactly storing strategies and reconstructing them later. However, it may miss out on opportunities for improvement. For example, if we apply Re-solve refinement to our base strategy in Coin Toss, we may arrive at the same strategy as the base strategy in which Player 2 chooses Forfeit 25% of the time, even though Heads and Tails dominate that action. The next endgame solving technique addresses this shortcoming.

Maxmargin Refinement

Maxmargin refinement (Moravcik et al. 2016) is similar to Re-solve refinement, except that it seeks to improve the endgame strategy as much as possible over the alternative payoff. While Re-solve refinement seeks a strategy for P_2 in S that would simply dissuade P_1 from entering S, Maxmargin refinement additionally seeks to punish P_1 as much as possible if P_1 nevertheless chooses to enter S. A *subgame margin* is defined for each infoset in S_r, which represents the difference in value between entering the subgame versus choosing the alternative payoff. Specifically, for each infoset $I \in S_r$ and action a_S leading to S, the *subgame margin* $M(I, a_S) = v^{\sigma^S}(I, a'_T) - v^{\sigma^S}(I, a'_S)$, or equivalently $M(I, a_S) = CBV^{\sigma_{-1}}(I, a) - v^{\sigma^S}(I, a'_S)$. In Maxmargin refinement, a Nash equilibrium strategy is computed such that the minimum margin over all $I \in S_r$ is maximized.

Given our base strategy in Coin Toss, Maxmargin refinement would result in P_2 choosing Heads with probability $\frac{3}{8}$, Tails with probability $\frac{5}{8}$, and Forfeit with probability 0.

Maxmargin refinement is safe. Furthermore, it guarantees that if every Player 1 best response reaches the endgame with positive probability through some infoset(s) that have positive margin, then exploitability is strictly lower than that of the base strategy.

Still, none of the prior techniques consider that in Coin Toss P_1 can achieve a payoff of 0.5 by choosing Left with Heads, and thus has more incentive to reach S when in the Tails state. The next section introduces our new technique, Reach-Maxmargin refinement, which solves this problem.

Reach-Maxmargin Refinement

In this section we introduce *Reach-Maxmargin refinement*, a new method for refining endgames that considers *what payoffs are achievable from other paths in the game*. We first consider the case of refining a *single* endgame in a game tree. We then cover independently refining multiple endgames.

Refining a Single Endgame

All of the endgame-solving techniques described in the previous section only consider the target endgame in isolation. This can be improved by incorporating information about what payoffs the players could receive by not reaching the endgame. For example in Coin Toss (Figure 1), P_1 can receive payoff 0.5 by choosing Left in the Heads state, and -0.5 in the Tails state. The solution that Maxmargin refinement produces would result in P_1 receiving payoff $-\frac{1}{4}$ by choosing Right in the Heads state, and $\frac{1}{4}$ in the Tails state. Thus, P_1 could simply always choose Left in the Heads state and Right in the Tails state against P_2's strategy and receive expected payoff $\frac{3}{8}$. Reach-Maxmargin improves upon this.

The auxiliary game used in Reach-Maxmargin refinement requires additional definitions. Define the *path* $Q_S(I)$ to an infoset $I \in S_r$ to be the set of infosets I' such that $I' \sqsubseteq I$ and I' is not an ancestor of any other information set in S_r. We also define $CBR_1(\sigma'_{-1})^{\to I \cdot a'_S}$ as the P_1 strategy that plays to reach $I \cdot a'_S$ in all infosets $I' \sqsubseteq I$, and elsewhere plays identically to $CBR_1(\sigma'_{-1})$.

We now describe the auxiliary game used in Reach-Maxmargin. The auxiliary game begins with a chance node that leads to $h' \in I'$ in proportion to $\pi^{\sigma}_{-1}(h')$, where I' is the earliest infoset such that $I' \in Q_S(I)$ for some $I \in S_r$. P_1 then has a choice between actions a'_T and a'_S. Action a'_T in Reach-Maxmargin refinement leads to a terminal payoff of $CBV^{\sigma_{-1}}(I')$. P_1 can instead take action a'_S, which can be viewed as P_1 attempting to reach $I \cdot a_S$ from I'. Since there may be P_2 nodes and chance nodes between I' and I, P_1 may not reach I from I' with probability 1. If P_1 reaches an infoset $I'' \notin Q_S(I)$ that is "off the path" from I, then we assume P_1 plays according to a counterfactual best response from that point forward and receives a payoff of $CBV^{\sigma_{-1}}(I'')$. However, with probability $\pi^{\sigma}_{-1}(h', h)$, P_1 can reach history $h \cdot a'_S$ for $h \in I$. From this point on, the auxiliary game is identical to that in Re-solve and Maxmargin refinement.

Formally, let σ' be the strategy that plays according to σ^S in S and otherwise plays according to σ. For an infoset $I \in S_r$ and action a_S leading to S, let I' be the earliest infoset such that $I' \sqsubseteq I$ and I' cannot reach an infoset in S_r other than I. We define a *reach margin* as

$$M_r(I, \sigma, \sigma_S) = CBV^{\sigma_{-1}}(I') - CBV^{\sigma'_{-1} \to I \cdot a'_S}(I')$$

Reach-Maxmargin refinement finds a Nash equilibrium σ^S in the auxiliary game such that the minimum margin $\min_I M_r(I, \sigma_S, S)$ is maximized. Theorem 1 shows that Reach-Maxmargin refinement results in a combined strategy with exploitability lower than or equal to the base strategy. If the opponent reaches a refined endgame with positive probability and the margin of the reached infoset is positive, then exploitability is *strictly lower* than that of the base strategy. This theorem statement is similar to that of Maxmargin refinement (Moravcik et al. 2016), but the margins here are higher than (or equal to) those in Maxmargin refinement.

Theorem 1. *Given a strategy σ_2, an endgame S for P_2, and a refined endgame Nash equilibrium strategy σ_2^S, let σ'_2 be the strategy that plays according to σ_2^S in endgame S and σ_2 elsewhere. If $\min_I M_r(I, \sigma, \sigma_S) \geq 0$ for S, then $exp(\sigma'_2) \leq exp(\sigma_2)$. Furthermore, if $\pi^{\langle BR^{\sigma'_2}, \sigma'_2 \rangle}(I) > 0$ for some $I \in S_r$ for an endgame S, then $exp(\sigma'_2) \leq exp(\sigma_2) - \pi^{\sigma'_2}_{-1}(I) \min_I M(I, \sigma'_2, S)$.*

The auxiliary game can be solved in a way that maximizes the minimum margin by using a standard LP solver. In order to use iterative algorithms such as the Excessive Gap Technique (Nesterov 2005; Gilpin, Peña, and Sandholm 2012) or Counterfactual Regret Minimization (CFR) (Zinkevich et al. 2007), one can use the *gadget game* described by Moravcik et al. (2016). Details on the gadget game are provided in the Appendix. In our experiments we used CFR.

Refining Multiple Endgames Independently

Other endgame solving methods have also considered the cost of reaching an endgame (Waugh, Bard, and Bowling 2009; Jackson 2014). However, those approaches (and the version of Reach-Maxmargin refinement we described above) are only correct in theory when applied to a *single* endgame. Typically, we want to refine multiple endgames independently—or, equivalently, any endgame that is reached at run time. This poses a problem because the construction of the auxiliary game assumes that all P_2 nodes outside the endgame have strategies that are fixed according to the base strategy. If this assumption is violated by refining multiple endgames, then the theoretical guarantees of Reach-Maxmargin refinement no longer hold.

To address this issue, we first add a constraint that $CBV^{\sigma'_{-1}}(I) \leq CBV^{\sigma_{-1}}(I)$ for every P_1 infoset. This trivially guarantees that $exp(\sigma'_2) \leq exp(\sigma_2)$. We also modify the Reach-Maxmargin auxiliary game. Let σ' be the strategy profile after all endgames are solved and recombined. Ideally, when solving an endgame S we would like any P_1 action leading away from S (that is, any action a belonging to an infoset $I' \in Q_S(I)$ such that $I' \cdot a \notin Q_S(I) \cup S$) to lead to a terminal payoff of $CBV_1^{\sigma'}(h \cdot a)$ rather than $CBV_1^{\sigma}(h \cdot a)$. However, since we are solving the endgames independently, we do not know what σ' will be. Nevertheless, we can have $h \cdot a$ lead to a *lower bound* on $CBV_1^{\sigma'}(h \cdot a)$. In our experiments we use the minimum reachable payoff as a lower bound.[1] Tighter upper and lower bounds, or accurate estimates of $CBV_1^{\sigma'}(I)$ for an infoset I, may lead to even better empirical performance.

Theorem 2 shows that even though the endgames are solved independently, if an endgame has positive minimum margin and is reached with positive probability then the final strategy will have lower exploitability than without Reach-Maxmargin endgame solving on that endgame.

Theorem 2. *Given a strategy σ_2, a set of disjoint endgames $\mathbb{S}$ for P_2, and a refined endgame Nash equilibrium strategy σ_2^S for each endgame $S \in \mathbb{S}$, let σ'_2 be the strategy that plays according to σ_2^S in each endgame S, respectively, and σ_2 elsewhere. Moreover, let σ_2^{-S} be the strategy that plays according to σ'_2 everywhere except for P_2 nodes in S, where it instead plays according to σ_2. If $\pi^{\langle BR^{\sigma'_2}, \sigma'_2 \rangle}(I) > 0$ for some $I \in S_r$, then $exp(\sigma'_2) \leq exp(\sigma_2^{-S}) - \pi_{-1}^{\sigma'_2}(I) \min_I M(I, \sigma_2^S, S)$.*

We now introduce an improvement to Reach-Maxmargin refinement. Let I' be an infoset in $Q_S(I)$. Let a_O be an action leading away from S and let a_Q be an action leading toward S. If the lower bound for $CBV^{\sigma'_S}(I', a_O)$ is higher than $CBV^{\sigma_S}(I', a_Q)$ then S will never be reached through I' in a Nash equilibrium. Thus, there is no point in further increasing the margin of I. This allows other margins to be larger instead, leading to better overall performance. This applies even when refining multiple endgames independently. We use this improvement in our experiments.

Nested Endgame Solving

As we have discussed, large games must be abstracted to reduce the game to a tractable size. This is particularly common in games with large or continuous action spaces. Typically the action space is discretized by action abstraction so only a few actions are included in the abstraction. While we might limit ourselves to the actions we included in the abstraction, an opponent might choose actions that are not in the abstraction. In that case, the off-tree action can be mapped to an action that is in the abstraction, and the strategy from that in-abstraction action can be used. This is certainly problematic if the two actions are very different, but in many cases it leads to reasonable performance. For example, in an auction game we might include a bid of \$100 in our abstraction. If a player bids \$101, we can probably treat that as a bid of \$100 without major problems. This is referred to as *action translation* (Gilpin, Sandholm, and Sørensen 2008; Schnizlein, Bowling, and Szafron 2009; Ganzfried and Sandholm 2013). Action translation is the state-of-the-art prior approach to dealing with this issue. It is used, for example, by all the leading competitors in the Annual Computer Poker Competition (ACPC). The leading action translation mapping—i.e., way of mapping opponent's off-tree actions back to actions in the abstraction—is the pseudoharmonic mapping (Ganzfried and Sandholm 2013); it has an axiomatic foundation, plays intuitively correctly in small sanity-check games, and is used by most of the leading teams in the ACPC. That is the action mapping that we will benchmark against in our experiments.

In this section, we develop techniques for applying endgame solving to calculate responses to opponent's off-tree actions, thereby obviating the need for action translation. We present two methods that dramatically outperform the leading action translation technique. The same techniques can also be used more generally to calculate finer-grained card or action abstractions as play progresses down the game tree. In this section, for exposition, we assume that P_2 wishes to respond to P_1 choosing an off-tree action.

The first method, which we refer to as the *inexpensive* method, begins by calculating a Nash equilibrium σ within the abstraction, and calculating $CBV^{\sigma_{-1}}(I, a)$ for each infoset $I \in \mathcal{I}_1$ and action a in the abstraction. When P_1 chooses an off-tree action a in infoset I, an endgame S is generated such that $I \in S_r$ and $I \cdot a$ leads to S. This endgame may be an abstraction. S is solved using any of the safe endgame solving techniques discussed earlier, except that we use $CBV^{\sigma_{-1}}(I)$ in place of $CBV^{\sigma_{-1}}(I, a)$ (since a is not a valid action in I according to σ). The solution σ^S is combined with σ to form σ'. $CBV^{\sigma'_{-1}}(I', a)$ is then calculated for each infoset $I' \in S$ and each $I' \in Q_S(I)$ (that is, on the path to I). The process repeats whenever P_1 again chooses an off-tree action in S.

By using $CBV^{\sigma_{-1}}(I)$ in place of $CBV^{\sigma'_{-1}}(I', a)$, we can retain some of the theoretical guarantees of Reach-Maxmargin refinement and Maxmargin refinement. Intuitively, if in every information set I P_1 is better off taking an action already in the game than the new action that was added, then the refined strategy is still a Nash equilibrium. Specifically, if the minimum reach margin $M_{\min}$ of the added action is nonnegative, then the combined strategy σ' is a Nash equilibrium in the expanded game that contains the new action. If $M_{\min}$ is negative, then the distance of σ' from a Nash equilibrium is proportional to $-M_{\min}$.

This "inexpensive" approach does not apply with Unsafe endgame solving because the probability of reaching an action outside of a player's abstraction is undefined. That is, $\pi^{\sigma}(h \cdot a)$ is undefined when a is not considered a valid action in h according to the abstraction. Nevertheless, a similar but more expensive approach is possible with Unsafe endgame solving (as well as all the other endgame-solving

[1] While this may seem like a loose lower bound, there are many situations where the off-path action simply leads to a terminal node. For these cases, the lower bound we use is optimal.

techniques) by starting the endgame solving at h rather than at $h \cdot a$. In other words, if action a taken in history h is not in the abstraction, then Unsafe endgame solving is conducted in the smallest endgame containing h (and action a is added to that abstraction). This increases the size of the endgame compared to the inexpensive method because a strategy must be recomputed for every action $a' \in A(h)$ in addition to a. For example, if an off-tree action is chosen by the opponent as the first action in the game, then the strategy for the entire game must be recomputed. We therefore refer to this method as the *expensive* method. We present experiments with both methods.

Experiments

We conducted our experiments on a poker game we call *No-Limit Flop Hold'em* (NLFH). NLFH is similar to the popular poker game of No-Limit Texas Hold'em except that there are only two rounds, called the pre-flop and flop. At the beginning of the game, each player receives two private cards from a 52-card deck. Player 1 puts in the "big blind" of 100 chips, and Player 2 puts in the "small blind" of 50 chips. A round of betting then proceeds starting with Player 2, referred to as the *preflop*, in which an unlimited number of bets or raises are allowed so long as a player does not put more than 20,000 chips (i.e., her entire chip stack) in the pot. Either player may fold on their turn, in which case the game immediately ends and the other player wins the pot. After the first betting round is completed, three community cards are dealt out, and another round of betting is conducted (starting with Player 1), referred to as the *flop*. At the end of this round, both players form the best possible five-card poker hand using their two private cards and the three community cards. The player with the better hand wins the pot.

For equilibrium finding, we used a version of CFR called CFR+ (Tammelin et al. 2015) with the speed-improvement techniques introduced by Johanson et al. (2011). There is no randomness in our experiments.

Our first experiment compares the performance of unsafe, re-solve, maxmargin, and reach-maxmargin refinement when applied to information abstraction (which is card abstraction in the case of poker). Specifically, we solve NLFH with no information abstraction on the preflop. On the flop, there are 1,286,792 infosets for each betting sequence; the abstraction buckets them into 30,000 abstract ones (using a leading information abstraction algorithm (Ganzfried and Sandholm 2014)). We then apply endgame solving immediately after the preflop ends but before the flop community cards are dealt. We experiment with two versions of the game, one small and one large, which include only a few of the available actions in each infoset. The small game has 9 non-terminal betting sequences on the preflop and 48 on the flop. The large game has 30 on the preflop and 172 on the flop. Table 1 shows the performance of each technique. In all our experiments, exploitability is measured in the standard units used in this field: milli big blinds per hand (mbb/h).

	Small Game	Large Game
Base Strategy	9.128	4.141
Unsafe	0.5514	39.68
Resolve	8.120	3.626
Maxmargin	0.9362	0.6121
Reach-Maxmargin	0.8262	0.5496

Table 1: Exploitability (evaluated in the game with no information abstraction) of the endgame-solving techniques.

Despite lacking theoretical guarantees, Unsafe endgame solving outperformed the safe methods in the small game. However, it did substantially worse in the large game. This exemplifies its variability. Among the safe methods, our Reach-Maxmargin technique performed best on both games.

The second experiment evaluates nested endgame solving using the different endgame solving techniques, and compares them to action translation. In order to also evaluate action translation, in this experiment, we create an NLFH game that includes 3 bet sizes at every point in the game tree (0.5, 0.75, and 1.0 times the size of the pot); a player can also decide not to bet. Only one bet (i.e., no raises) is allowed on the preflop, and three bets are allowed on the flop. There is no information abstraction anywhere in the game. [2] We also created a second, smaller abstraction of the game in which there is still no information abstraction, but the 0.75x pot bet is never available. We calculate the exploitability of one player using the smaller abstraction, while the other player uses the larger abstraction. Whenever the large-abstraction player chooses a 0.75x pot bet, the small-abstraction player generates and solves an endgame for the remainder of the game (which again does not include any 0.75x pot bets) using the nested endgame solving techniques described above. This endgame strategy is then used as long as the large-abstraction player plays within the small abstraction, but if she chooses the 0.75x pot bet later again, then the endgame solving is used again, and so on. Table 2 shows that all the endgame solving techniques substantially outperform action translation. Resolve, Maxmargin, and Reach-Maxmargin use inexpensive nested endgame solving, while Unsafe and "Reach-Maxmargin (expensive)" use the expensive approach. Reach-Maxmargin refinement performed the best, outperforming maxmargin refinement and unsafe endgame solving. These results suggest that nested endgame solving is preferable to action translation (if there is sufficient time to solve the endgame).

Conclusion

We introduced an endgame solving technique for imperfect-information games that has stronger theoretical guarantees

[2] There are no chip stacks in this version of NLFH. Chip stacks pose a considerable challenge to action translation, because the optimal strategy in a poker game can change drastically when any player has bet almost all her chips. Since action translation maps each bet size to a bet size in the abstraction, it may significantly overestimate or underestimate the number of chips in the pot, and therefore perform extremely poorly when near the chip stack limit. Refinement techniques do not suffer from the same problem. Conducting the experiments without chip stacks is thus conservative in that it favors action translation over the endgame solving techniques. We nevertheless show that the latter yield significantly better strategies.

	Exploitability
Randomized Pseudo-Harmonic Mapping	146.5
Resolve	15.02
Reach-Maxmargin (Expensive)	14.92
Unsafe (Expensive)	14.83
Maxmargin	12.20
Reach-Maxmargin	11.91

Table 2: Comparison of the various endgame solving techniques in nested endgame solving. The performance of the pseudo-harmonic action translation is also shown. Exploitability is evaluated in the large action abstraction, and there is no information abstraction in this experiment.

and better practical performance than prior endgame-solving methods. We presented results on exploitability of both safe and unsafe endgame solving techniques. We also introduced a method for nested endgame solving in response to the opponent's off-tree actions, and demonstrated that this leads to dramatically better performance than the usual approach of action translation. This is, to our knowledge, the first time that exploitability of endgame solving techniques has been measured in large games.

Acknowledgments

This material is based on work supported by the NSF under grants IIS-1617590, IIS-1320620, and IIS-1546752, the ARO under award W911NF-16-1-0061.

References

Bellman, R. 1965. On the application of dynamic programming to the determination of optimal play in chess and checkers. *Proceedings of the National Academy of Sciences* 53(2):244–246.

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 18th International Joint Conference on Artificial Intelligence (IJCAI)*.

Burch, N.; Johanson, M.; and Bowling, M. 2014. Solving imperfect information games using decomposition. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Ganzfried, S., and Sandholm, T. 2013. Action translation in extensive-form games with large action spaces: Axioms, paradoxes, and the pseudo-harmonic mapping. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Ganzfried, S., and Sandholm, T. 2014. Potential-aware imperfect-recall abstraction with earth mover's distance in imperfect-information games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Ganzfried, S., and Sandholm, T. 2015. Endgame solving in large imperfect-information games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Gilpin, A., and Sandholm, T. 2006. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1007–1013.

Gilpin, A., and Sandholm, T. 2007. Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em poker. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 1168–1175.

Gilpin, A.; Peña, J.; and Sandholm, T. 2012. First-order algorithm with $\mathcal{O}(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games. *Mathematical Programming* 133(1–2):279–298. Conference version appeared in AAAI-08.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2008. A heads-up no-limit Texas Hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Jackson, E. 2014. A time and space efficient algorithm for approximately solving large imperfect information games. In *AAAI Workshop on Computer Poker and Imperfect Information*.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Johanson, M. 2013. Measuring the size of large no-limit poker games. Technical report, University of Alberta.

Moravcik, M.; Schmid, M.; Ha, K.; Hladik, M.; and Gaukrodger, S. J. 2016. Refining subgames in large imperfect information games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Nash, J. 1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences* 36:48–49.

Nesterov, Y. 2005. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal of Optimization* 16(1):235–249.

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

Schaeffer, J.; Burch, N.; Björnsson, Y.; Kishimoto, A.; Müller, M.; Lake, R.; Lu, P.; and Sutphen, S. 2007. Checkers is solved. *Science* 317(5844):1518–1522.

Schnizlein, D.; Bowling, M.; and Szafron, D. 2009. Probabilistic state translation in extensive games with large action sets. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*.

Tammelin, O.; Burch, N.; Johanson, M.; and Bowling, M. 2015. Solving heads-up limit Texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.

Waugh, K.; Bard, N.; and Bowling, M. 2009. Strategy grafting in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Zinkevich, M.; Bowling, M.; Johanson, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Appendix: Supplementary Material

Description of Gadget Game

Solving the auxiliary game described in Maxmargin Refinement and Reach-Maxmargin Refinement will not, by itself, maximize the minimum margin. While LP solvers can easily handle this objective, the process is more difficult for iterative algorithms such as Counterfactual Regret Minimization (CFR) and the Excessive Gap Technique (EGT). For these iterative algorithms, the auxiliary game can be modified into a *gadget game* that, when solved, will provide a Nash equilibrium to the auxiliary game and will also maximize the minimum margin (Moravcik et al. 2016).

The gadget game differs from the auxiliary game in two ways. First, all P_1 payoffs that are reached from the initial information set of I' are shifted by $CBV^{\sigma_{-1}}(I', a)$ in Maxmargin refinement and by $CBV^{\sigma_{-1}}(I')$ in Reach-Maxmargin refinement. Second, rather than the game starting with a chance node that determines P_1's starting state, P_1 will get to decide for herself which state to begin the game in. Specifically, the game begins with a P_1 node where each action in the node corresponds to an information set I in S_r for Maxmargin refinement, or the earliest infoset $I' \in Q_S(I)$ for Reach-Maxmargin refinement. After P_1 chooses to enter an information set I, chance chooses the precise history $h \in I$ in proportion to $\pi_{-1}^{\sigma_{-1}}(h)$.

By shifting all payoffs by $CBV^{\sigma_{-1}}(I', a)$ or $CBV^{\sigma_{-1}}(I')$, the gadget game forces P_1 to focus on improving the performance of each information set over some baseline, which is the goal of Maxmargin and Reach-Maxmargin refinement. Moreover, by allowing P_1 to choose the state in which to enter the game, the gadget game forces P_2 to focus on maximizing the minimum margin.

Figure 4 illustrates the gadget game for Maxmargin refinement.

Proof of Theorem 1

Proof. Assume $M_r(I, \sigma, \sigma_S) \geq 0$ for every information set I in S_r for an endgame S and let $\epsilon = \min_I M_r(I, \sigma, \sigma_S)$.

For an information set $I \in S_r$, let I' be the earliest information set in $Q_S(I)$. Then $CBV^{\sigma_{-1}}(I') \geq CBV^{\sigma'_{-i} \to I \cdot a'_S}(I') + \epsilon$.

First suppose that $\pi^{\langle BR(\sigma'_2), \sigma'_2 \rangle}(I) = 0$. Then either $\pi^{\langle BR(\sigma'_2), \sigma'_2 \rangle}(I') = 0$ or $\pi^{\langle BR(\sigma'_2), \sigma'_2 \rangle}(I', I) = 0$. If it is the former case, then $CBV^{\sigma_{-1}}(I')$ does not affect $exp(\sigma'_2)$. If it is the latter case, then since I is the only information set in S_r reachable from I', so in any best response I' only reaches nodes outside of S with positive probability. The nodes outside S belonging to P_2 were unchanged between σ and σ', so $CBV^{\sigma'_{-1}}(I') \leq CBV^{\sigma_{-1}}(I')$.

Now suppose that $\pi^{\langle BR(\sigma'_2), \sigma'_2 \rangle}(I) > 0$. Since $BR(\sigma'_2)$ already reaches I' on its own, so $CBV^{\sigma'_{-i}}(I') = CBV^{\sigma'_{-i} \to I \cdot a'_S}(I')$. Since $CBV^{\sigma_{-1}}(I') \geq CBV^{\sigma'_{-i} \to I \cdot a'_S}(I') + \epsilon$, so we get $CBV^{\sigma_{-1}}(I') \geq CBV^{\sigma'_{-i}}(I') + \epsilon$. This is the condition for Theorem 1 in Moravcik et al. (2016). Thus, from that theorem, we get that $exp(\sigma'_2) \leq exp(\sigma_2) - \epsilon \pi_{-1}^{\sigma'_2}(I)$.

Now consider any information set $I'' \sqsubset I'$. Before encountering any P_2 nodes whose strategies are different in σ' (that is, P_2 nodes in S), P_1 must first traverse a I' information set as previously defined. But for every I' information set, $CBV^{\sigma'_{-1}}(I') \leq CBV^{\sigma_{-1}}(I')$. Therefore, $CBV^{\sigma'_{-1}}(I'') \leq CBV^{\sigma_{-1}}(I'')$. □

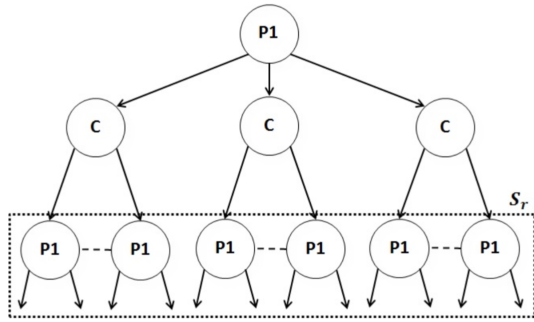

Figure 4: An example of a gadget game in Maxmargin refinement. P_1 picks the initial information set she wishes to enter S_r in. Chance then picks the particular history of the information set, and play then proceeds identically to the auxiliary game. All P_1 payoffs are shifted by $CBV^{\sigma_{-1}}(I', a)$.

Proof of Theorem 2

Proof. Let $S \in \mathcal{S}$ be an endgame for P_2 and assume $\pi^{\langle BR^{\sigma'_2}, \sigma'_2 \rangle}(I) > 0$ for some $I \in S_r$. Let $\epsilon = \min_I M_r(I, \sigma, \sigma_S)$ and let I' be the earliest information set in $Q_S(I)$. Since we added the constraint that $CBR^{\sigma'_{-1}}(I) \leq CBR^{\sigma_{-1}}(I)$ for all P_1 information sets, so $\epsilon \geq 0$. We only consider the non-trivial case where $\epsilon > 0$. Since $BR(\sigma'_2)$ already reaches I' on its own, so $CBV^{\sigma'_{-i}}(I') = CBV^{\sigma'_{-i} \to I \cdot a'_S}(I')$.

Let $\sigma_2'^S$ represent the strategy which plays according to σ_2^S in P_2 nodes of S and elsewhere plays according to σ. Since $\epsilon > 0$ and we assumed the minimum payoff for every P_1 action in $Q_S(I)$ that does not lead to I, so $CBV^{\sigma'^S_{-1} \to I \cdot a'_S}(I') \leq BRV^{\sigma^{-S}_{-1}}(I') - \epsilon$.

Moreover, since σ'^S_{-1} assumes a value of $CBV^{\sigma_{-1}}(h)$ is received whenever a history $h \notin Q_S(I)$ is reached due to chance or P_2, and $CBV^{\sigma_{-1}}(h)$ is an upper bound on $CBV^{\sigma'_{-1}}(h)$, so $CBV^{\sigma'^S_{-1} \to I \cdot a'_S}(I') \geq CBV^{\sigma'_{-1} \to I \cdot a'_S}(I')$.

Thus, $CBV^{\sigma'_{-1} \to I \cdot a'_S}(I') \leq BRV^{\sigma^{-S}_{-1}}(I') - \epsilon$. Finally, since I' can be reached with probability $\pi^{\sigma_{-1}}(I')$, so $exp(\sigma'_2) \leq exp(\sigma_2^{-S}) - \pi_{-1}^{\sigma'_2}(I) \min_I M(I, \sigma_2^S, S)$.

□