A Time and Space Efficient Algorithm for Approximately Solving Large Imperfect Information Games

Eric Jackson

Abstract

This paper proposes a novel approach for computing an approximate equilibrium in a game of imperfect information. Our approach involves decomposition; that is, breaking the problem down into subproblems that can be solved independently. We compare our approach to other decomposition approaches, and illustrate that our approach is guaranteed to find an equilibrium when applied to the recovery problem.

Introduction

This paper proposes a novel approach for computing an approximate equilibrium in a game of imperfect information. Solving very large games of imperfect information requires large amounts of both memory and computation. Generally, the memory requirement is proportional to the number of information sets in the game being solved. While researchers typically solve abstractions that are smaller than the full game, it is advantageous for the abstractions to be as large as possible so as to best approximate the full game. Some labs employ supercomputers with 256 GB of RAM or more to solve these large games.

Within the field of computer poker, many researchers have proposed decomposition approaches (e.g., (Billings et al. 2003), (Burch, Johanson, and Bowling 2014), (Ganzfried and Sandholm 2013), (Gilpin and Sandholm 2006), (Waugh, Bard, and Bowling 2009)) that break the problem down into pieces that can be solved independently. If we can decompose a game into enough pieces, we should be able to solve each subproblem on commodity hardware with only a (relatively) modest amount of RAM. While the total amount of computation required may not be less, we can employ a distributed network of commodity hardware rather than one large supercomputer. This will avoid a hard limit on the amount of computation we can apply over a fixed amount of time, as we can always add more machines, and at the same time may be a more cost-effective solution.

Unfortunately, games of imperfect information do not easily lend themselves to decompounding. In this respect, they differ from games of perfect information like chess. In chess, it is possible to take a certain state of the board and identify the most promising move based on that current state, independent of any other possible state of the game. However, this is not typically possible in games of imperfect information. See (Burch, Johanson, and Bowling 2014) for more discussion.

Many decomposition approaches attempt to identify an approximate Nash equilibrium, but lack theoretical guarantees that the solutions the algorithms produce will converge to equilibrium. (See (Burch, Johanson, and Bowling 2014) for an exception.) The algorithm we propose is ultimately no different in this respect, although we do show that in a certain "best-case" scenario, convergence to an equilibrium is guaranteed.

While our algorithm is not specific to any one game-solving approach, we illustrate how it would work in conjunction with counterfactual regret minimization (CFR) (Zinkevich et al. 2007). Throughout we are concerned only with two-person zero-sum games with perfect recall. Examples are given involving two-player variants of poker, specifically Texas Hold'em.

Extensive Games

We begin by recapping the definition of an extensive game. For a fuller discussion, see (Osborne and Rubenstein 1994).

Definition 1. *An **extensive game** Γ has the following components:*

- *A finite set N of players*
- *A finite set H of histories of actions. $Z \subseteq H$ are the terminal histories. $A(h) = \{a : (h, a) \in H\}$ are the actions available after a nonterminal history h.*
- *A player function P that assigns to each non-terminal history a member of $N \cup c$ where c represents chance. H_i is the set of histories where player i chooses the next action.*
- *A function f_c that associates with every history $h \in H_c$ a probability distribution over the actions $a \in A(h)$.*
- *For each player $i \in N$, a utility function u_i that assigns a utility for that player to each terminal history. Because we are dealing only with two-player zero-sum games $u_1(z) = -u_2(z)$. It will be possible therefore just to have a single utility function u which we can assume, without loss of generality, to specify player 1's utility.*
- *For each player $i \in N$, a partition $\boldsymbol{I}_i$ of H_i with the property that $A(h) = A(h')$ whenever h and h' are in the same*

member of the partition. A set $I_i \in \boldsymbol{I}_i$ is an information set of player i.

We'll write $h \sqsubset h'$ if h is a prefix of h'. And similarly $h \sqsubseteq h'$ if $h = h'$ or $h \sqsubset h'$.

Definition 2. *A **strategy** for player i, σ_i, assigns a probability distribution over $A(h)$ to each $h \in H_i$.*

Definition 3. *A **strategy profile** σ is a set of strategies $\{\sigma_1, \ldots \sigma_n\}$ that contains one strategy for each player.*

When all players play according to a strategy profile σ we refer to the expected utility of player 1 as $u(\sigma)$.

Letting $\sigma(h, a)$ be the probability assigned to a at h by σ, we can define the joint probability of a history for σ:

$$\pi^\sigma(h) = \Pi_{(h',a) \sqsubseteq h} \sigma(h', a) \tag{1}$$

We'll also use π_i^σ to refer to the product of only the terms where player i acts, and π_{-i}^σ to refer to the product of only the terms where players other than i act. We can also define $\pi^\sigma(h, h')$ where $h \sqsubseteq h'$ to be the product of only the terms between h and h'.

Let V be the value of Γ from player 1's perspective; i.e., $u(\sigma)$ for any equilibrium strategy profile σ. The Minimax theorem tells us that all equilibrium strategy profiles in a two-player zero-sum game yield the same expected payoff to each player. Throughout we state expected payoffs from player 1's perspective meaning that player 1 plays to maximize the expected payoff, and player 2 plays to minimize the expected payoff.

Recovery Problem

The problem of recovering a strategy in a subtree of a game is discussed in (Burch, Johanson, and Bowling 2014) where it is a key part of the CFR-D algorithm. We discuss the recovery problem here as a benchmark for comparing different approaches to decomposition. For this task, we imagine that we are given an equilibrium strategy profile for a game, except that all information for a certain subtree has been lost. We are tasked with "recovering" an equilibrium for the full game.

It will not typically be possible to guarantee recovery of exactly the same equilibrium. The object is just to recover *any* equilibrium to the full game. So long as we are dealing with two-player zero-sum games, we know (due to the Minimax theorem) that any equilibrium for the full game yields the same expected payoff to each player, so in that sense at least all equilibria are equivalent.

An approach to decomposition that allows us to solve subtrees independently ought to be applicable to the recovery problem. One way to evaluate these approaches is based on whether they have any theoretical guarantees with respect to the recovery game — do they guarantee finding an equilibrium for the full game? We can also evaluate their efficiency on the recovery game. Are their time and space requirements proportional to the size of the missing subtree?

Approaches to Decomposition

Standard Approach

One common and natural approach to decomposition can be found in different forms in multiple papers including (Billings et al. 2003), (Ganzfried and Sandholm 2013) and (Gilpin and Sandholm 2006). Because it recurs so often, we are referring to it here as the "standard" approach. The standard approach operates in two passes: in the first pass you solve a coarse abstraction of the full game; in the second pass you go back and resolve one or more subtrees using a finer abstraction. To resolve a subtree, we fix the strategies outside of the subtree to be whatever was computed in the first pass, and we then compute new strategies for the subtree, each player's strategy within the subtree being allowed to vary freely. With fixed strategies outside of the subtree, it is not necessary to run a solver over the whole tree. We can instead compute the distribution of information sets at the root of the subtree, and then simply solve the subtree as an independent game, with that particular distribution of information sets at the root.

The standard approach has several nice properties. We get the advantages of a finer abstraction in the resolved subtrees, and each resolve has memory and computation requirements only proportional to the size of the subtree.

The standard approach is an example of what we'll call a "two pass" approach. In a two pass approach, you typically solve the whole game on a first pass, and then go back and resolve one or more subtrees on a second pass. Normally, there is a change of abstraction between the first pass and the second pass. On the first pass, we might solve the whole game with a coarse abstraction, while the resolves on the second pass use a finer abstraction. While the standard approach is one example of a two pass approach, it is not the only one, and, indeed, the other decomposition approaches described in this paper are two pass approaches.

It is well known that when using the standard approach, there is no guarantee that you will find an equilibrium for the full game, or even an approximate equilibrium. This is true even if there is no change of abstraction between the first and second pass. In other words, the standard approach has no guarantees on the recovery game. (Ganzfried and Sandholm 2013) illustrates this with the game of rock-paper-scissors and shows that you cannot recover the second player's equilibrium strategy even given the first player's strategy. For this example, imagine we have computed the equilibrium for rock-paper scissors, which is that both players select rock, paper and scissors each with probability 1/3. We then discard the second player's strategy and try to recover it using the standard approach. In this example, player 2's actions form the subtree and player 1's actions are outside of the subtree. Any best-response by player 2 forms an equilibrium in the endgame (since player 1's strategy is fixed) and any distribution over rock, paper and scissors is equally good against player 1's fixed strategy. So the standard approach has no ability to distinguish the unique equilibrium strategy $\langle 1/3, 1/3, 1/3 \rangle$ from any other player 2 strategy.

Strategy Grafting

An alternative approach to decomposition is described in (Waugh, Bard, and Bowling 2009) and has more theoretical guarantees than the standard approach. Strategy grafting is a two pass approach: the entire game is solved using a coarse abstraction, and then subtrees are resolved using a finer abstraction. On the second pass, subtree strategies are computed separately for each player. The key idea is that when we solve for player 1, we fix player 1's strategy outside of the subtree, but let the opponent's strategy vary over the whole tree. Any solving approach can be used, including but not limited to CFR. The key contrast between strategy grafting and the standard approach is that we fix only one player's strategy outside of the subtree as opposed to both.

While the recovery problem is not specifically discussed in (Waugh, Bard, and Bowling 2009), we can still analyze how this algorithm would work when applied to that problem.

First, we formalize what we mean by "subtree". A set T is a subtree for a game Γ if:

1. $T \subseteq H$
2. If $h \in T$ and h is a prefix of h', then $h' \in T$
3. For each player i if $h \in T$ and there exists I_i in $\mathbf{I}_i$ such that $h \in I_i$ and $h' \in I_i$, then $h' \in T$.

Note that such a subset T might better be called a forest than a tree because there are multiple roots corresponding to different histories in the same information set.

In discussing the recovery problem, it is important to distinguish between different games. The full game denoted by Γ_f is the game of interest that we want to recover an equilibrium for. There are also recovery games that we solve to obtain the strategy for the subtree of interest. With strategy grafting, there are two recovery games, which we'll denote as $\Gamma_{r,1}$ and $\Gamma_{r,2}$ because we recover the subtree strategy separately for each player. The recovery game $\Gamma_{r,i}$ is derived from the full game Γ_f, given a subtree T and an equilibrium strategy profile σ_f for Γ_f. It differs in that for all $h \in H_{f,i} \setminus T$, $P(h) = c$ and $f_c(a|h) = \sigma_{f,i}(h, a)$. In other words, at all histories outside of T, player i actions are replaced by chance actions with the probabilities specified by the given full-game equilibrium strategy.

Solving the recovery game $\Gamma_{r,i}$ yields the equilibrium strategy profile $\{\sigma_{r,i,1}, \sigma_{r,i,2}\}$. We define the expansion of $\sigma_{r,i,i}$, denoted $\sigma^+_{r,i,i}$, to be the full game strategy that is identical to $\sigma_{r,i,i}$ on histories in T and identical to $\sigma_{f,i}$ elsewhere.

Our first theorem states that $\langle \sigma^+_{r,1,1}, \sigma^+_{r,2,2} \rangle$ is an equilibrium for Γ_f. In other words, strategy grafting solves the recovery problem. A proof can be found in the first appendix.

We can view the strategy grafting approach as forcing the target player to learn a more robust strategy than the standard approach. The opponent has the freedom to play different hands to the subtree, so the target player is forced to develop a robust strategy that performs reasonably well no matter what the opponent does.

The memory requirements for strategy grafting are reduced from solving the full game, but still substantial. For example, in a CFR implementation, outside of the subtree, we would need to maintain regrets for the opponent, but not the accumulated strategy. We would not need to maintain regrets or the accumulated strategy for the target player outside of the subtree, although we would need to maintain the fixed already computed strategy. Similarly, the computation requirements are reduced but still substantial because we would need to update the strategy for the opponent outside of the subtree on each iteration.

Strategy grafting contrasts with the standard approach in that with the standard approach the memory and computation requirements are proportional only to the size of the subtree, whereas with strategy grafting they are proportional to the size of the full tree. In practice, the subtree is likely to often be much smaller than the full tree, so this may be a very substantial difference.

Our Approach

Application To The Recovery Problem

The approach we propose for decomposition can be viewed as an extension of strategy grafting. We begin by describing how we would use this approach to solve the recovery problem, and then generalize in the subsequent section.

For motivation, note that the strategy grafting algorithm spends a great deal of time computing the strategy for the opponent outside of the subtree. This may seem wasteful as a) we do not care about the result of that computation; we are only computing a strategy for the target player within the subtree, and b) since the target player has a fixed strategy outside of the subtree, it seems that we should be able to do something simpler or faster than a full equilibrium-finding approach which is designed for the more general scenario where both players can vary their strategies.

As in the earlier discussion, we denote the "full" game of interest with Γ_f, and assume an equilibrium strategy profile σ_f has been previously computed for that game. A portion of σ_f corresponding to a subtree has been lost and we want to recover an equilibrium for the full game. Just as with strategy grafting, we will be resolving for each player separately, so we have two recovery games $\Gamma_{r,1}$ and $\Gamma_{r,2}$. However, our recovery games are smaller; we eliminate almost all game states that are not in the subtree. We maintain the subtree itself, all states on the path to the subtree, and have new terminal histories after each action that takes us off the path to the subtree. This is depicted in figure 1. The new terminal histories have payoffs corresponding to the value the opponent would achieve if play deviates from the path to the subtree (assuming both players play according to σ_f).

When we solve the recovery game for player i we will let player i's strategy vary just over the subtree T, but the opponent's strategy can vary over the whole recovery game.

Define the path P_T to a subtree T as:

$$P_T = \{h : h \notin T \;\&\; \exists h' : h' \in T \;\&\; h \sqsubset h'\} \qquad (2)$$

Also define the set O_T of "off-path" histories from Γ_f:

$$O_T = \{h : h \notin T \cup P_T \;\&\; \exists h' : h = (h', a) \;\&\; h' \in P_T\} \qquad (3)$$

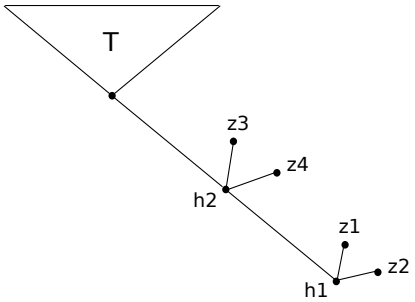

Figure 1: Depiction of the subtree T, the path to the subtree $\{h1, h2\}$ and newly created terminal nodes $z1 \dots z4$.

Let $Z(h)$ be the terminal histories reachable from h:

$$Z(h) = \{z \in Z : h \sqsubseteq z\} \tag{4}$$

We can now define the values that will be used for the new off-path terminal histories:

$$\upsilon_\sigma(h) = \sum_{z \in Z(h)} \pi^\sigma(h, z) u(z) \tag{5}$$

$\upsilon_\sigma(h)$ is the expected utility at h assuming each player plays to reach h.

Now we can define the recovery game $\Gamma_{r,i}$ which is derived from Γ_f. Let $j = 3 - i$; i.e., j is the index of the opponent of player i. The set of histories is $T \cup P_T \cup O_T$. The utility function $u_{r,i}$ is identical to u_f except that for $h \in O_T$ we set $u_{r,i}(h) = \upsilon_\sigma(h)$. Also, for all $h \in P_T \cap H_i$, $P(h) = c$ and $f_c(a|h) = \sigma_{f,i}(h, a)$.

Solving the recovery game $\Gamma_{r,i}$ yields the equilibrium strategy profile $\{\sigma_{r,i,1}, \sigma_{r,i,2}\}$. We define the expansion $\sigma^+_{r,i,i}$ as before. Our second theorem states that $\left\{\sigma^+_{r,1,1}, \sigma^+_{r,2,2}\right\}$ is an equilibrium for Γ_f. In other words, the approach we have described here solves the recovery problem. A proof can be found in the second appendix.

Decomposition

The task we are ultimately interested in is not the recovery problem, but the computation of a strategy for a game from scratch using decomposition. To do this, we adopt a two pass approach. We solve the full game of interest with a coarse abstraction, and then resolve subtrees with a finer abstraction. The algorithm for resolving subtrees is just as described above. Unfortunately, once we allow a change of abstractions between the solving of the full game and the solving of the subtrees, we lose the optimality results that obtained for the recovery problem. Consider that we will be computing the off-path expected values based on the coarse abstraction, but there is no guarantee that those are the correct values for the fine abstraction. Indeed, they are almost certainly not identical to the values that would obtain had we solved the full game with the fine abstraction.

Having said that, it seems possible that the values obtained from the first pass will be similar to the values we would have obtained had we solved the full game with the finer abstraction. Also, even if we compute slighly incorrect values on the first pass, it seems entirely possible that we may still be able to learn good strategies for the subtrees on the second pass.

The fact that our algorithm has equilibrium guarantees for the recovery game may lend us some confidence that it finds robust strategies for the subtrees, even though we are lacking theoretical guarantees in the two-pass decomposition scenario.

When we solve a subtree we are solving a recovery game that is approximately the size of the subtree, with a small number of additional game states along the path to the subtree. Typically, the number of game states along the path to the subtree is dwarfed by the size of the subtree. The memory and computation requirements for our approach are thus proportional to the size of the subtree, with the computation requirements being roughly double the requirements of the standard approach. (Recall that we compute strategies for each of the two players separately whereas the standard approach does not need to.)

Comparison to CFR-D

CFR-D, described in (Burch, Johanson, and Bowling 2014), is an algorithm for solving games of imperfect information via decomposition. It has certain resemblances to the algorithm described here. CFR-D finds an equilibrium for a large game with memory requirements only proportional to the size of largest "piece" that the game is decomposed into. CFR-D operates in two passes. The first pass in some sense solves the whole game, but throws away information about each subtree after processing it on each iteration, only maintaining information about the "trunk". (This is the key property that keeps the memory requirement so small.) The second pass recovers the equilibrium for the full game based on results of the first pass, by solving lots of recovery problems.

CFR-D computes counterfactual values at the root of subtrees on the first pass, and uses those in the second pass to guide the solutions of the recovery problems. These counterfactual values serve an analogous role to the values we compute for the off-path terminal nodes. Both approaches work in the sense that they can solve the recovery problem. But in the context of a two-pass approach with a change of abstraction, they are not equivalent.

Although we can think of CFR-D as a two-pass algorithm, there is no change of abstraction between the first pass and the second pass. Partly because CFR-D uses the same abstraction throughout, it is possible to prove that CFR-D actually finds an exact equilibrium for the full game. In contrast, the two-pass approaches with a change of abstraction discussed in this paper have no such theoretical guarantees.

Practical application of CFR-D to large games is likely constrained today by the large amount of computation required on the first pass.

Results

We have applied the standard approach and our approach to the game of Leduc. Leduc is a small variant of poker played with a deck of six cards. There are two rounds of betting, a

maximum of two bets per street and a limit betting structure. The results were as follows.

Game	Exploitability (mbb/g)
Whole game	0.49
Standard approach	38.53
Our approach	0.93

The whole game and each of the recovery games were solved with 100 million iterations of pure external CFR. Exploitability numbers are expressed in milli-big-blinds per game (mbb/g). As you can see, the approach described in this paper finds a solution much closer to an equilibrium than the standard approach.

Implementation

CFR Implementation

The exposition above is agnostic about the solving approach used to compute equilibria to games. In our implementation, we have used counterfactual regret minimization (CFR). Certain adjustments can make for a more efficient CFR implementation. For example, in our definition of the recovery games, we defined values $v(h)$ which became payoffs at the new off-path terminal nodes. For CFR, what we will typically want is the counterfactual value of an *information set*. There are many fewer information sets than histories so it will be advantageous to precompute just the needed values - counterfactual values of information sets at off-path terminal nodes.

These counterfactual values could be computed by a single iteration of Vanilla CFR. However, even a single iteration of Vanilla CFR is quite expensive for very large games. We instead use a single pass of CFR in which we sample both from the public cards (the boards) and the opponent's actions. This produces estimates of the needed values, rather than exact values.

Imperfect Recall

The results above obtain only for games with perfect recall. However, the approach can be applied to games of imperfect recall with empirically reasonable results. One requirement, however, is that the off-path counterfactual values be "history-sensitive", so to speak. It is not sufficient to compute them for an imperfect recall bucket. They could be computed for specific card combinations, or for a sequence of imperfect recall buckets.

Slumbot 2014

The decomposition approach described in this paper is being employed in the ongoing development of Slumbot 2014, our entry to this year's Annual Computer Poker Competition. We have built a base system and are in the process of resolving some of the most commonly reached flop subtrees. The base strategy will continue to be employed for the preflop strategy and for less commonly reached flop subtrees.

We are employing machines with 32 gigabytes of RAM. The base system and each of the subtrees are solved on a dedicated machine using all available RAM. We are using a variant of pure external CFR, which requires storage of two four-byte integers for each action at each information set, implying that the size of each game is approximately four billion information set / action pairs.

Appendix One: Strategy Grafting Proof

Let V_f be the value of Γ_f (i.e., $u(\sigma_f)$). Recall that we state expected payoffs from player 1's perspective meaning that player 1 plays to maximize the expected payoff, and player 2 plays to minimize the expected payoff.

Definition 4. *$\tau_i^- \in \Sigma_{r,i,i}$ is the **restriction** of a full-game strategy τ_i for player i if τ_i^- is identical to τ_i on all histories in T.*

Definition 5. *Given an equilibrium strategy profile σ_f for Γ_f and given a strategy $\tau_i \in \Sigma_{r,i,i}$, τ_i^+ is the **expansion** of τ_i. It is identical to τ_i on histories in T and identical to $\sigma_{f,i}$ elsewhere.*

Note that the restriction of the expansion of any strategy τ_i in $\Sigma_{r,i,i}$ is τ_i.

Lemma 1. *$u_{r,i}(\{\tau_{r,i,i}, \tau_{f,j}\}) = u_f(\{\tau^+_{r,i,i}, \tau_{f,j}\})$ for any $\tau_{r,i,i} \in \Sigma_{r,i,i}$, $\tau_{f,j} \in \Sigma_{f,j}$, $i \in \{1,2\}$, $j = 3 - i$.*

Proof. Recall that:

$$u(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u(z) \tag{6}$$

Observe that $\pi^{\{\tau_{r,i,i}, \tau_{f,j}\}}(z) = \pi^{\left\{\tau^+_{r,i,i}, \tau_{f,j}\right\}}(z)$ for all terminal histories z. Additionally, $u(z)$ is the same in $\Gamma_{r,i}$ and Γ_f for all z. □

Henceforth we will use σ_f to refer to the given equilibrium strategy profile for the full game. $\sigma^-_{f,i}$ is the restriction of $\sigma_{f,i}$.

Theorem 1. *$\left\{\sigma^-_{f,i}, \sigma_{f,j}\right\}$ is an equilibrium for $\Gamma_{r,i}$ with value V_f for $i \in \{1,2\}$, $j = 3 - i$.*

Proof. We'll show this for player 1 ($i = 1$). The proof for player 2 is exactly the same.

First we show that the best response to $\sigma^-_{f,1}$ in $\Gamma_{r,1}$ is V_f. The best response to $\sigma^-_{f,1}$ is:

$$min_{\tau \in \Sigma_{r,1,2}} u(\left\{\sigma^-_{f,1}, \tau\right\}) \tag{7}$$

Since $\Sigma_{r,1,2} = \Sigma_{f,2}$:

$$= min_{\tau \in \Sigma_{f,2}} u(\left\{\sigma^-_{f,1}, \tau\right\}) \tag{8}$$

By lemma 1:

$$= min_{\tau \in \Sigma_{f,2}} u(\{\sigma_{f,1}, \tau\}) = V_f \tag{9}$$

So the best response to $\sigma^-_{f,1}$ is V_f.

We also show that the best response to $\sigma_{f,2}$ in $\Gamma_{r,1}$ is $\leq V_f$. Suppose there were a strategy $\tau \in \Sigma_{r,1,1}$ such that:

$$u(\{\tau, \sigma_{f,2}\} = V' > V_f \tag{10}$$

By the lemma:

$$u(\{\tau^+, \sigma_{f,2}\} = V' > V_f \tag{11}$$

which is a contradiction.

These two conclusions together signify that $\left\{\sigma^-_{f,1}, \sigma_{f,2}\right\}$ is an equilibrium for $\Gamma_{r,1}$ with value V_f. The proof for $\Gamma_{r,2}$ is exactly the same.

□

For the following corollary, assume we solve $\Gamma_{r,1}$ and $\Gamma_{r,2}$, and find two equilibrium strategy profiles $\{\sigma_{r,1,1}, \sigma_{r,1,2}\}$ and $\{\sigma_{r,2,1}, \sigma_{r,2,2}\}$. These two equilibria may be different from $\left\{\sigma^-_{f,i}, \sigma_{f,j}\right\}$ and $\left\{\sigma_{f,i}, \sigma^-_{f,j}\right\}$, but their values must still be V_f due to the Minimax theorem.

Corollary 1. *$\{\sigma^+_{r,1,1}, \sigma^+_{r,2,2}\}$ is an equilibrium for Γ_f with value V_f.*

Proof. The best response to $\sigma_{r,i,i}$ in $\Gamma_{r,i}$ has been shown to be V_f. Since $\Sigma_{r,i,j} = \Sigma_{f,j}$ (player j — the opponent — has the same available responses in $\Gamma_{r,i}$ and Γ_f), and due to lemma 1, we know that the best response to $\sigma^+_{r,i,i}$ in Γ_f is also V_f. □

Appendix Two: Proof for Our Approach

We use the Greek letter σ to denote equilibrium strategies. In particular, for the recovery problem we assume we are given an equilibrium strategy profile σ_f for the full game Γ_f. We'll also use $\sigma_{r,i}$ to refer to an equilibrium strategy profile in the recovery game $\Gamma_{r,i}$ and $\sigma_{r,i,j}$ to refer to the player j strategy in such a profile. If we wish to refer to an arbitrary strategy or strategy profile we use the Greek letter τ.

We define expansions and restrictions similarly to before. The restriction of a strategy τ_i, τ_i^-, is τ_i restricted to histories in the given recovery game. Given a full game strategy profile σ_f, the expansion of τ_i, τ_i^+, is identical to τ_i on histories in the recovery game and identical to $\sigma_{f,i}$ elsewhere.

If h is a history, we can write (h, a) to denote the history following h after action a is taken. We say that (h, a) is a child of h in such a case. We previously defined $h \sqsubset h'$ and $h \sqsubseteq h'$ when h is a (strict) prefix of h' and we can also say that h' is a descendant of h when $h \sqsubset h'$.

Recall that an information set I is a set of histories. We can also say that an information set I' is a child of I if every history $h' \in I'$ is a child of some history $h \in I$. Similarly we can talk about an information set I' being a descendant of I.

Let $Z(I)$ be the set of terminal histories $z \in Z$ such that $h \sqsubset z$ for some $h \in I$. Likewise $Z(I, a)$ is the set of terminal histories $z \in Z$ such that $(h, a) \sqsubseteq z$ for some $h \in I$

Let $z[I]$ be the longest history $h \in I$ such that $h \sqsubseteq z$.

We define the counterfactual value of an action a at an information set I under a given strategy profile τ:

Definition 6.

$$v(\tau, I, a) = \sum_{z \in Z(I,a)} \pi^\tau_i((z[I], a), z)\pi^\tau_{-i}(z)u(z) \tag{12}$$

Similarly we can define the counterfactual value of an information set I under a given strategy profile τ:

Definition 7.

$$v(\tau, I) = \sum_{z \in Z(I)} \pi^\tau_i(z[I], z)\pi^\tau_{-i}(z)u(z) \tag{13}$$

Counterfactual values and utilities are stated from the perspective of player 1. Player 1 plays to maximize utility, and player 2 plays to minimize it.

We'll also say that an information set I is "reachable" with respect to a strategy profile τ if $\pi^\tau(I) > 0$.

Lemma 2. *For any strategy profile τ for the recovery game $\Gamma_{r,i}$, $u_r(\tau) = u_f(\tau^+)$*

We use u_r and u_f to be explicit whether we are talking about utilities in the recovery game or the full game.

$$u_r(\tau) \tag{14}$$

$$= \sum_{z \in Z_r} \pi^\tau(z)u_r(z) \tag{15}$$

$$= \sum_{z \in T \cap Z_r} \pi^\tau(z)u_r(z) + \sum_{z \in O_T} \pi^\tau(z)u_r(z) \tag{16}$$

$$= \sum_{z \in T \cap Z_r} \pi^\tau(z)u_r(z) + \sum_{z_1 \in O_T} \pi^\tau(z_1) \sum_{z_2 \in Z_f \,\&\, z_1 \sqsubseteq z_2} \pi^{\sigma_f}(z_1, z_2)u_f(z_2) \tag{17}$$

$$= \sum_{z \in T \cap Z_r} \pi^\tau(z)u_r(z) + \sum_{z_1 \in O_T} \pi^\tau(z_1) \sum_{z_2 \in Z_f \,\&\, z_1 \sqsubseteq z_2} \pi^{\tau^+}(z_1, z_2)u_f(z_2) \tag{18}$$

$$= \sum_{z \in T \cap Z_f} \pi^{\tau^+}(z)u_f(z) + \sum_{z \in Z_f \,\&\, z \notin T} \pi^{\tau^+}(z)u_f(z) \tag{19}$$

$$= \sum_{z \in Z_f} \pi^{\tau^+}(z)u_f(z) \tag{20}$$

$$= u_f(\tau^+) \tag{21}$$

From lemma 2, we know in particular:

$$u_r(\left\{\sigma^-_{f,1}, \sigma^-_{f,2}\right\}) = u_f(\{\sigma_{f,1}, \sigma_{f,2})\}) = V_f \tag{22}$$

Lemma 3. *In $\Gamma_{r,i}$, $\left\{\sigma^-_{f,1}, \sigma^-_{f,2}\right\}$ is an equilibrium profile.*

Proof. Take $\Gamma_{r,1}$. Suppose there were a strategy $\tau_2 \in \Sigma_{r,1,2}$ such that $u(\sigma^-_{f,1}, \tau_2) = V' < V_f$. From lemma 2, we then know $u(\sigma_{f,1}, \tau_2^+) = V' < V_f$. This is a contradiction as it shows $\sigma_{f,2}$ is not a best response to $\sigma_{f,1}$.

Similarly, suppose there were a strategy $\tau_1 \in \Sigma_{r,1,1}$ such that $u(\tau_1, \sigma^-_{f,2}) = V' > V_f$. From lemma 2, we then know $u(\tau_1^+, \sigma_{f,2}) = V' > V_f$. This is a contradiction as it shows $\sigma_{f,1}$ is not a best response to $\sigma_{f,2}$.

The proof for $\Gamma_{r,2}$ is exactly the same. □

If we solve $\Gamma_{r,i}$ we will get an equilibrium $\{\sigma_{r,i,1}, \sigma_{r,i,2}\}$ which also has value V_f due to the Minimax theorem. Moreover, $\left\{\sigma_{r,i,i}, \sigma^-_{f,j}\right\}$ (where $j = 3 - i$) must also be an equilibrium for $\Gamma_{r,i}$. Since $u(\sigma_{r,i,i}, \sigma^-_{f,j}) = V_f$, we also know $u(\sigma^+_{r,i,i}, \sigma_{f,j}) = V_f$ by lemma 2.

We can now show that we have a method for recovering an equilibrium for the full game:

Theorem 2.

$$\left\{\sigma^+_{r,1,1}, \sigma^+_{r,2,2}\right\} \textit{ is an equilibrium for } \Gamma_f \tag{23}$$

Proof. We show that $\sigma^+_{r,1,1}$ is player 1's half of an equilibrium. The proof for player 2 is exactly the same.

We show that $\sigma_{f,2}$ is a best response to $\sigma^+_{r,1,1}$. We know from above that $u(\sigma^+_{r,1,1}, \sigma_{f,2}) = V_f$, so if $\sigma_{f,2}$ is moreover a best response to $\sigma^+_{r,1,1}$, then we have shown that $\sigma^+_{r,1,1}$ must be player 1's half of an equilibrium.

We prove a small lemma first:

Lemma 4. *Suppose 1) $\{\sigma_1, \sigma_2\}$ is an equilibrium; and 2) I is an information set reachable in $\{\sigma_1, \sigma_2\}$. Consider any strategy τ_2 such that $v(\{\sigma_1, \tau_2\}, I, a) \geq v(\{\sigma_1, \sigma_2\}, I, a)$ for all actions a available at I. Then: $v(\{\sigma_1, \tau_2\}, I) \geq v(\{\sigma_1, \sigma_2\}, I)$.*

Proof. If I is an information set at which player 1 or chance acts, then the lemma is trivially true because the probability distribution over actions will be identical between $\{\sigma_1, \sigma_2\}$ and $\{\sigma_1, \tau_2\}$. So assume player 2 acts at I. Since $\{\sigma_1, \sigma_2\}$ is an equilibrium and I is reachable, player 2 only selects actions with non-zero probability that minimize the counterfactual value:

$$v(\{\sigma_1, \sigma_2\}, I) = min_a v(\{\sigma_1, \sigma_2\}, I, a) \tag{24}$$

Since none of the counterfactual values of the actions a when adopting τ_2 can be lower than the corresponding values for σ_2 by assumption, τ_2 cannot achieve a lower counterfactual value at I.

□

Lemma 4 essentially says that an equilibrium strategy σ_2 acts in a locally optimal fashion at any reachable information set I.

To prove that $\sigma_{f,2}$ is a best response to $\sigma^+_{r,1,1}$, we will show that the following property holds for any $\tau_2 \in \Sigma_{f,2}$ and any information set I that is reachable in $\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}$:

$$v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, I) \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I) \tag{25}$$

In other words, there is no player 2 strategy τ_2 that achieves better counterfactual value than $\sigma_{f,2}$ at any reachable information set.

Suppose this weren't true. Then there would be an information set I^* which violates the property (25). Choose I^* such that no descendant I' of I^* violates the property.

Observe that for all a available at I^*:

$$v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, I, a) \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I, a) \tag{26}$$

There are two possibilities. If a leads to a terminal history z, then the statement holds with equality because:

$$v(\left\{\sigma^+_{r,1,1}, \tau\right\}, I^*, a) = \pi^\tau_{-i}(z)u(z) \tag{27}$$

and neither $\pi^\tau_{-i}(z)$ nor $u(z)$ depend on the player 2 strategy.

The second possibility is that action a at I^* leads to a new information set I'. But by our assumption that no descendant violates the property, we know $v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, I') \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I')$ which is equivalent to $v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, I^*, a) \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I^*, a)$.

We now wish to show that the hypothesized information set I^* cannot exist because it contradicts lemma 4. First observe that the following holds for all $I \in T \cup P_T$:

$$v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I, a) = v(\left\{\sigma_{r,1,1}, \sigma^-_{f,2}\right\}, I, a) \tag{28}$$

and also:

$$v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I) = v(\left\{\sigma_{r,1,1}, \sigma^-_{f,2}\right\}, I) \tag{29}$$

Similarly, for all $I \notin T \cup P_T$:

$$v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I, a) = v(\{\sigma_{f,1}, \sigma_{f,2}\}, I, a) \tag{30}$$

and also:

$$v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I) = v(\{\sigma_{f,1}, \sigma_{f,2}\}, I) \tag{31}$$

Therefore, if the hypothesized I^* is in $T \cup P_T$, we have a contradiction of lemma 4 with respect to the recovery game equilibrium profile $\left\{\sigma_{r,1,1}, \sigma^-_{f,2}\right\}$. On the other hand, if I^* is not in $T \cup P_T$, we have a contradiction of lemma 4 with respect to the full game equilibrium $\{\sigma_{f,1}, \sigma_{f,2}\}$.

Since we have reached a contradiction, we know that

$$v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, I) \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, I) \tag{32}$$

for all reachable information sets I. In particular, we know this for the information set at the root of the game, $\varnothing$:

$$v(\left\{\sigma^+_{r,1,1}, \tau_2\right\}, \varnothing) \geq v(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}, \varnothing) \tag{33}$$

But the counterfactual value at $\varnothing$ is identical to expected utility ($v(\sigma, \varnothing) = u(\sigma)$) so we have shown for all $\tau_2 \in \Sigma_{f,2}$:

$$u(\left\{\sigma^+_{r,1,1}, \tau_2\right\}) \geq u(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}) \tag{34}$$

In other words, $\sigma_{f,2}$ is a best response to $\sigma^+_{r,1,1}$. It was previously shown that:

$$u(\left\{\sigma^+_{r,1,1}, \sigma_{f,2}\right\}) = V_f \tag{35}$$

So $\sigma^+_{r,1,1}$ is player 1's half of an equilibrium in the full game, Γ_f.

□

References

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 2003 International Joint Conference on Artificial Intelligence (IJCAI)*.

Burch, N.; Johanson, M.; and Bowling, M. 2014. Solving imperfect information games using decomposition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-14)*.

Ganzfried, S., and Sandholm, T. 2013. Improving performance in imperfect-information games with large state and action spaces by solving endgames. In *Computer Poker and Imperfect Information Workshop at the National Conference on Artificial Intelligence (AAAI-13)*.

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-06)*.

Osborne, M. J., and Rubenstein, A. 1994. *A Course in Game Theory*. The MIT Press.

Waugh, K.; Bard, N.; and Bowling, M. 2009. Strategy grafting in extensive games. In *Advances in Neural Information Processing Systems 22 (NIPS)*.

Zinkevich, M.; Bowling, M.; Johanson, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20 (NIPS)*.