Refining Subgames in Large Imperfect Information Games

Matej Moravcik, Martin Schmid, Karel Ha, Milan Hladik
Charles University In Prague
{moravcim, schmidm, karelha, hladik}
@kam.mff.cuni.cz,

Stephen J. Gaukrodger
Koypetition
stephen@koypetition.com

Abstract

The leading approach to solving large imperfect information games is to pre-calculate an approximate solution using a simplified abstraction of the full game; that solution is then used to play the original, full-scale game. The abstraction step is necessitated by the size of the game tree. However, as the original game progresses, the remaining portion of the tree (the subgame) becomes smaller. An appealing idea is to use the simplified abstraction to play the early parts of the game and then, once the subgame becomes tractable, to calculate a solution using a finer-grained abstraction in real time, creating a combined final strategy. While this approach is straightforward for perfect information games, it is a much more complex problem for imperfect information games. If the subgame is solved locally, the opponent can alter his play in prior to this subgame to exploit our combined strategy. To prevent this, we introduce the notion of subgame margin, a simple value with appealing properties. If any best response reaches the subgame, the improvement of exploitability of the combined strategy is (at least) proportional to the subgame margin. This motivates subgame refinements resulting in large positive margins. Unfortunately, current techniques either neglect subgame margin (potentially leading to a large negative subgame margin and drastically more exploitable strategies), or guarantee only non-negative subgame margin (possibly producing the original, unrefined strategy, even if much stronger strategies are possible). Our technique remedies this problem by maximizing the subgame margin and is guaranteed to find the optimal solution. We evaluate our technique using one of the top participants of the AAAI-14 Computer Poker Competition, the leading playground for agents in imperfect information settings.

Introduction

Extensive form games are a powerful model capturing a wide class of real-world problems. The games can be either perfect information (Chess) or imperfect information (poker). Applications of imperfect information games range from security problems (Pita et al. 2009) to card games (Bowling et al. 2015)

The largest imperfect information game to be (essentially) solved today is the limit version of two-player Texas Hold'em poker (Bowling et al. 2015), with approximately 10^{17} nodes (Johanson 2013). Unfortunately, many games remain that are much too large to be solved with current techniques. For example, the more popular "No-Limit" variant of two-player Texas Hold'em poker has approximately 10^{165} nodes (Johanson 2013).

The leading approach to solving imperfect information games of this magnitude is to create a simplified abstraction of the game, compute an ϵ-equilibrium in the abstract game, and finally use the strategy from the abstracted game to play the original, unabstracted game (Billings et al. 2003) (Sandholm 2010) (Johanson et al. 2013) (Gibson 2014). The amount of simplification needed to produce the abstracted game is determined by the maximum size of the game tree that we are able to learn with the computing resources available. While abstraction pathologies mean that larger abstractions are not guaranteed to produce better strategies (Waugh et al. 2009), empirical results have shown that finer-grained abstractions are generally better (Johanson et al. 2013)

An appealing compromise is to pre-calculate the largest possible abstraction we can handle for the entire game and then improve this in real-time with refinements. The original strategy is used to play the early parts of the game (the trunk) and once the remaining portion of the game tree (the subgame) becomes tractable, we can refine the strategy for the subgame in real-time using even finer-grained abstraction. Figure 1 illustrates the approach.

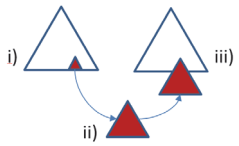

Figure 1: Subgame refinement framework. (i) the strategy for the game is pre-computed using coarse-grained abstraction (ii) during the play, once we reach a node defining a sufficiently small subgame, we refine the strategy for that subgame (iii) this together with the original strategy for the trunk creates a combined strategy. The point is to produce improved combined strategy

Note that not only can we enlarge the size of the abstraction in the subgame, we can also reduce the "off the tree problem". When an opponent takes an action that is not

found in the abstraction, it needs to be mapped onto a (similar) one in the abstraction. This mapping can destroy relevant game information. To reduce this effect, we can construct the subgame so that it starts in the exact state of the game so far (Ganzfried and Sandholm 2015).

Subgame refinement has been successfully used in perfect information games to improve the strategies (Müller and Gasser 1996) (Müller 2002). Unfortunately, the nature of imperfect information games means that it is difficult to isolate subgames. Current attempts to apply subgame refinement to imperfect information games have lead to marginal gains or potentially result in a more exploitable final solution. The reason for this is that if we change our strategy in the subgame then this gives our opponent the opportunity to exploit our combined strategy by altering their behavior in the trunk of the game. See (Burch, Johanson, and Bowling 2013) or (Ganzfried and Sandholm 2015) for details and several nice examples of this flaw.

The first approach, "endgame solving", does not guarantee a decrease in exploitability, and can instead produce a strategy that is drastically more exploitable. (Ganzfried and Sandholm 2015). The second approach, re-solving, was originally designed for subgame strategy re-solving. In other words, it aims to reproduce the original strategy from a compact representation. The resulting strategy is guaranteed to be no more exploitable than the original one. Although this technique can be used to refine the subgame strategy, there is no explicit construction that forces the refined strategy to be any better than the original, even if much stronger strategies exist. (Burch, Johanson, and Bowling 2013)

In this paper, we present a new technique, max-margin subgame refinement, that is tailor-made to reduce exploitability in imperfect information games. We introduce the notion of subgame margin, a simple value with appealing properties, which motivates subgame refinements that result in large positive margins.

We regard the problem of safe subgame refinement as a linear optimization problem. This perspective demonstrates the drawbacks and connections between the two previous approaches, and ultimately introduce linear optimization to maximize the subgame margin. Subsequently, we describe an imperfect information game construction that can be used to find such a strategy (rather than solving the resulting linear optimization problem). This allows us to solve larger subgames using recently introduced techniques, namely the CFR+ (Tammelin et al. 2015) and domain-specific speedup tricks (Johanson et al. 2012).

Finally, we experimentally evaluate all the approaches - endgame solving, re-solving and max-margin subgame refinement. For the first time, we evaluate these techniques on the safe-refinement task as part of a large-scale game by using one of the top participating agents in AAAI-14 Computer Poker Competition as the baseline strategy to be refined in subgames.

Previous Work

Despite the lack of theoretical guarantees, variants of subgame refinement have been used in imperfect information games for some time. The poker agent GS1-G4 (Gilpin and Sandholm 2006) (Gilpin, Sandholm, and Sørensen 2007) and its successor Tartanian (Ganzfried and Sandholm 2013) (Ganzfried and Sandholm 2015) used various techniques to either refine or solve the endgame. The authors call their newest version of their approach "endgame solving", and report both positive practical performance results as well as potentially negative impacts on the exploitability of the combined strategy (Ganzfried and Sandholm 2015). This is a property shared by all of these variants - the resulting strategy can be substantially more exploitable than the original strategy started with.

We are aware of only one prior subgame refinement technique that is guaranteed to produce a combined strategy that is no-more exploitable than the original strategy, re-solving (Burch, Johanson, and Bowling 2013) The technique works by computing the best response values for the opponent and using these values to construct a gadget game. Unfortunately, there is no explicit mechanism to cause the refined strategy to be any better than the original one, even if much stronger strategies are possible. By formulating this technique as an optimization problem, we can easily see this property.

Background and Notation

An extensive form game (Osborne and Rubinstein 1994, p. 200) consists of (i) A finite set of **players** P. (ii) A finite set H of all possible game states. Each member of H is a **history**, each component of history is an **action**. (iii) The empty sequence is in H, and every prefix of a history is also history ($(h,a) \in H \implies (h \in H)$). $h \sqsubseteq h'$ denotes that h is a prefix of h'. $Z \subseteq H$ are the terminal histories (they are not a prefix of any other history). (iv) The set of actions available after every non-terminal history $A(h) = \{a : (h,a) \in H\}$. (v) A function p that assigns to each non-terminal history an **acting player** (member of $P \cup c$, where c stands for chance). (vi) A function f_c that associates with every history for which $p(h) = c$ a probability measure on $A(h)$. Each such probability measure is independent of every other such measure. (vii) For each player $i \in P$, a partition $\mathcal{I}_i$ of $h \in H : p(h) = i$. $\mathcal{I}_i$ is the **information partition** of player i, with property that $A(h) = A(h')$ whenever h and h' are in the same member of the partition. A set $I_i \in \mathcal{I}_i$ is an **information set** of player i and we denote by $A(I_i)$ the set $A(h)$ and by $P(I_i)$ the player $P(h)$ for any $h \in I_i$ (viii) For each player $i \in P$ an **utility function** $u_i : Z \to \mathbb{R}$.

In the rest of the paper, we assume that the game is **perfect recall**, two-player **zero sum**. This means $P = \{1,2\}$, $u_1(z) = -u_2(z)$ and no player forgets any information revealed to him (nor the order it was revealed in).

A **strategy** for player i, σ_i, is a function that maps $I \in \mathcal{I}_i$ to a probability distribution over $A(I)$ and $\pi^\sigma(I,a)$ is the probability of action a. Σ_i denotes the set of all strategies of player i. A **strategy profile** is a vector of strategies of all players, $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{|P|})$. Σ denotes the set of all strategy profiles.

We denote $\pi^\sigma(h)$ as the probability of history h occurring given the strategy profile σ. Let $\pi_i^\sigma(h)$ be the contribution of player i to that probability. We can then decompose

$\pi^\sigma(h)$ as $\pi^\sigma(h) = \prod_{i \in P \cup c} \pi_i^\sigma(h)$. Let $\pi_{-i}^\sigma(h)$ be the product of all players contribution (including chance), except that of player i. For $I \in \mathcal{I}$, $\pi^\sigma(I) = \sum_{h \in I} \pi^\sigma(h)$ is the probability of reaching particular information set given σ and $\pi_p^\sigma(I)$, $\pi_{-p}^\sigma(I)$ again denote the player's contribution to this probability. We use $\pi^\sigma(h'|h)$ to refer to the probability of going from history h to the history h'.

Define $\sigma|_{I \to a}$ to be the same strategy profile as σ, except that a player always plays the action a in the information set I. Define $u_i(\sigma)$ to be the expected utility for player i, given the strategic profile σ, in other words $u_i(\sigma) = \sum_{h \in Z} u_i(h)\pi^\sigma(h)$.

A **Nash equilibrium** is a strategy profile σ such that for every player $i \in P$: $u_i(\sigma) \geq \max_{\sigma_i^* \in \Sigma_i} u_i((\sigma_i^*, \sigma_{-i}))$

The **Counterfactual value** $v_p^\sigma(I)$ is the expected utility given that information set I is reached and all players play using strategy σ, except that player p plays to reach I

$$v_p^\sigma(I) = \frac{\sum_{h \in I, h' \in Z} \pi_{-p}^\sigma(h)\pi^\sigma(h'|h)u_i(h')}{\pi_{-p}^\sigma(I)}$$

A **best response** $BR_p(\sigma)$ is a strategy of the player p that maximizes his expected utility given σ_{-p}.

In a two-player zero-sum game, the **exploitability** refers to strategy's additional loss to a best response compared to player's utility in a Nash equilibrium.

A **counterfactual best response** $CBR_p(\sigma)$ is a strategy where $\sigma_p(I, a) > 0$ iff $v_p^{\sigma|_{I \to a}}(I) = max_{a'} v_p^{\sigma|_{I \to a'}}(I)$. It maximizes counterfactual value at every information set. CBR_p is a always a best response but best response may not be contractual best response since it can choose an arbitrary action in information sets where $\pi_p(I) = 0$.

The well-known recursive tree walk algorithm for best response computation produces a counterfactual best response.

To simplify the notation we define a **counterfactual best response value** $CBV_p^\sigma(I)$. It is very similar to standard definition of counterfactual value, with exception that player p plays according to $CBR_p(\sigma)$ instead of σ. Formally $CBV_p^\sigma(I) = v_p^{(\sigma_{-p}, CRB_p(\sigma))}(I)$

Subgame

In a perfect information game, a subgame is a subtree of the original game tree rooted at any node. This definition is problematic for imperfect information games, since such subtree could include one part of an information set and exclude another. To define a subgame for an imperfect information game, a generalized concept of information set is used. Information set $I(h)$ groups histories that the acting player $p = P(h)$ cannot distinguish. **Augmented information set** set adds also histories that any of the remaining players cannot distinguish (Burch, Johanson, and Bowling 2013). Using this notion, one can define subgame.

Definition 1. *An imperfect information subgame (Burch, Johanson, and Bowling 2013) is a forest of trees, closed under both the descendant relation and membership within augmented information sets for any player.*

Note that root of the subgame, denoted $R(S)$, will not typically be a single (augmented) information set because different players typically have different information available to them, thus grouping of histories to augmented information sets will be different. We denote the set of all information sets of the player p at the root of the subgame as $\mathcal{I}_p^{R(S)}$.

Formulating Subgame Refinement using Optimization

In this section, we briefly describe the two current techniques - (i) endgame solving (Ganzfried and Sandholm 2015) and (ii) re-solving (Burch, Johanson, and Bowling 2013) We also reformulate both of them as equivalent optimization problems. Regarding these techniques as optimizations helps us to see the underlying properties of these two techniques. Subsequently, we use these insights to motivate our new, max-margin technique. We will assume, without loss of generality, that we are refining the strategy for player 1 (p_1) for the rest of this paper.

Endgame Solving

We start by constructing a fine-grained subgame abstraction. The original strategies for the subgame are discarded and only the strategies prior to the subgame (trunk) are needed. The strategies in the trunk are used to compute the joint distribution (belief) over the states at the beginning of the subgame. Finally, we add a chance node just before the fine-grained subgame. The node leads to the states at the root of the subgame. The chance node plays according to the computed belief. Adding the chance node roots the subgame, thus making it well-defined game. See Figure 2.

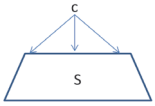

Figure 2: Endgame solving construction - Gadget 1. The (c)hance plays according to the belief computed using the trunk's strategy. The finer-grained (S)ubgame follows.

The following is a formulation of the linear optimization problem corresponding to the game construction. $LP1$ is the standard sequence form LP for the Gadget 1.

$$\begin{aligned} \max_{v,x} f^\top v \\ Ex &= e \\ F^\top v - A_1^\top x &\leq 0 \\ x &\geq 0 \end{aligned}$$

LP1 - optimization problem corresponding to endgame solving. A_1 is the sequence form payoff matrix, x is the vector of p_1 strategies, v is the vector of (negative) counterfactual best response values for p_2, E and F are sequence constraint matrices and e is sequence constraint vector (Nisan et al. 2007) (Čermák, Bošanskỳ, and Lisy 2014)

The flaw in this technique stems from the fact that even if the trunk strategy (and thus the starting distribution) is optimal, the combined strategy can become drastically more exploitable. (Ganzfried and Sandholm 2015) (Burch, Johanson, and Bowling 2013)

Re-solving

Again, we start by creating a fine-grained abstraction for the subgame. The original strategy for the subgame (from the coarse abstraction) is then translated into the fine-grained abstraction as σ_1^S. The translated strategy is now used to compute $CBV_2^{\sigma_1^S}(I)$ for every information set I at the root of the subgame. These values will be useful for the gadget construction to guarantee the safety of the resulting strategy.

To construct the gadget, we add one chance node at the root of the game, followed by additional nodes for p_2 - one for every state at the root of the subgame. At each of these nodes, p_2 may either accept the corresponding counterfactual best response value calculated earlier or play the subgame (to get to the corresponding state at the root of the subgame). The chance player distributes the p_2 into these states using the (normalized) π^{σ}_{-2} (how likely is the state given that p_2 plays to reach it). Since the game is zero sum, this forces p_1 to play the subgame well enough that the opponent's value is no greater than the original CBV. See Figure 3 for a sketch of the construction. For more details see (Burch, Johanson, and Bowling 2013).

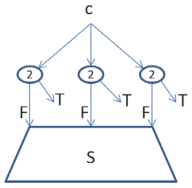

Figure 3: re-solving gadget construction - Gadget 2. The opponent chooses in every state prior to the endgame to either (F)ollow the action into the endgame or to (T)erminate. His utility after the (T)erminal action is set to his counterfactual best response in that state.

Next, we formulate a linear optimization problem corresponding to the gadget construction. This time, the presented LP is not a straightforward sequential-form representation of the construction. Although such a representation would be possible, it would not help provide the insight we are seeking. Instead, we formulate a LP that solves the same game (for the p_1) while demonstrating the underlying properties of the re-solving approach. The formulation uses the fact that any strategy for which the opponent's current counterfactual best response is no greater than the original one, is a solution to the game (this follows form the construction of Gadget 2).

$$\begin{aligned} \max_{v,x} \ & \mathbf{0} \\ v_I &\geq CBV_2^{\sigma}(I), \quad I \in \mathcal{I}_2^{R(S)} \\ Ex &= e \\ F^{\top}v - A_2^{\top}x &\leq 0 \\ x &\geq 0 \end{aligned}$$

LP2 - $\mathcal{I}_2^{R(S)}$ denotes the root information sets, $CBV_2^{\sigma}(I)$ is the original counterfactual best response value of p_2 in the information set I. The sequence payoff matrices A_1 and A_2 are slightly different to reflect different strategy of the chance player in Gadget 1 and Gadget 2.

It is worth noting three critical points here.

1. $LP2$ is not maximizing any value, but rather finding a feasible solution (though theoretically equivalent, it is semantically different for the strategy in this case).
2. The original, unrefined strategy is a solution to $LP2$
3. Although 1) and 2) suggest that the strategy might not improve, empirical evaluations show that if one uses a CFR algorithm to solve the corresponding game (Gadget 2), the refined strategy's performance improves upon the original(Burch, Johanson, and Bowling 2013). Our experiments further confirm this.

Discussion

Looking at the $LP1$ and $LP2$, it's easy to see the properties of existing approaches. The $LP1$ (endgame solving) lacks the constraints ($v_I \geq CBV_2^{\sigma}(I)$) that bound the exploitability, possibly producing strategy drastically more exploitable than the original one. $LP2$ (re-solving) bounds the exploitability, but lacks maximization factor, possibly producing strategies no better than the original one. As we will see, our approach both bounds the exploitability while maximizing some well-motivated function.

Our Technique

The outline of this section is following: 1. we list the steps used by our technique 2. we use the problem of refining imperfect information subgames to motivate a value to maximized 3. we formalize this value as the subgame margin 4. we discuss and formalize its properties 5. we formulate an LP optimizing the subgame margin 6. we describe a corresponding extensive form game construction - Gadget 3

Our technique follows the steps of the subgame refinement framework: (i) Create an abstraction for the game. (ii) Compute an equilibrium approximation within the abstraction. (iii) Play according to this strategy. (iv) When the play reaches final stage of the game, create a fine-grained abstraction for the endgame. (v) Refine the strategy in the fine-grained abstraction. (vi) Use the resulting strategy in that subgame (creating a combined strategy).

Since all the steps except of the step five are identical to already described techniques, we describe only this steps in details.

Subgame Margin

To address the potential increase in exploitability caused by an opponent altering his behavior in the trunk, we ensure that there is no distribution of starting states that would allow him to increase his CBV when confronted by subgame refinement. The simplest way to ensure this is to decrease his CBV in all possible starting states. We can put a lower bound on this improvement by measuring the state with the smallest decrease in CBV. Our goal is to maximize this lower bound. We refer to this values as the **subgame margin**.

Definition 2. *Subgame Margin*

Let σ_1, σ_1' be a pair of p_1 strategies for subgame S. Then a subgame margin

$$SM_1(\sigma_1, \sigma_1', S) = \min_{I_2 \in \mathcal{I}_2^{R(S)}} CBV_2^{\sigma_1}(I_2) - CBV_2^{\sigma_1'}(I_2)$$

Subgame margin has several useful properties. The exploitability is strongly related to the value of the margin. If it is non-negative, the new combined strategy is guaranteed to be no more exploitable than original one. Furthermore, given that the opponent's best response reaches the subgame with non-zero probability, the exploitability of our combined strategy is reduced. This improvement is at least proportional to the subgame margin (and may be greater).

Theorem 1. *Given a strategy σ_1, a subgame S and a refined subgame strategy σ_1^S, let $\sigma_1' = \sigma_1[S \leftarrow \sigma_1^S]$ be a combined strategy of σ_1 and σ_1^S. Let the subgame margin $SM_1(\sigma_1, \sigma_1', S)$ be non-negative. Then $u_1(\sigma_1', CBR(\sigma_1')) - u_1(\sigma_1, CBR(\sigma_1)) \geq 0$. Furthermore, if there is a best response strategy $\sigma_2^* = BR(\sigma_1')$ such that $\pi^{(\sigma_1', \sigma_2^*)}(I_2) > 0$ for some $I_2 \in \mathcal{I}_2^{R(S)}$, then $u_1(\sigma_1', CBR(\sigma_1')) - u_1(\sigma_1, CBR(\sigma_1)) \geq \pi_{-2}^{\sigma_1'}(I_2)\, SM_1(\sigma_1, \sigma_1', S)$.*

This theorem is generalization of the Theorem 1 *in (Burch, Johanson, and Bowling 2013). Intuitively, it follows from the way one computes a best response using the bottom-up algorithm. For the formal proof, see appendix A or the authors' homepage.*

Though this lower bound might seem artificial at first, it has promising properties for subgame refinement. Since we refine the strategy once we reach the subgame, we are either facing p_2's best response that reaches S or he has made a mistake earlier in the game. Furthermore, the probability of reaching a subgame is proportional to $\pi_{-2}^{\sigma_1'}(I_2)$. As this term (and by extension, the bound) increases, the probability of reaching that subgame grows. Thus, we are more likely to reach a subgame with larger bound.

Optimization Formulation

To find a strategy that maximizes the subgame margin, we can easily modify the $LP2$.

$$\begin{aligned} \max_{v,x}\ & m \\ v_I - m &\geq CBV_2^{\sigma}(I), \quad I \in \mathcal{I}_2^{R(S)} \\ Ex &= e \\ F^\top v - A_2^\top x &\leq 0 \\ x &\geq 0 \end{aligned}$$

LP3 - maximizing the subgame margin, m is scalar corresponding to the subgame margin that we aim to maximize.

The similarities between $LP3$ and $LP2$ make it easier to see that where the $LP2$ optimization guarantees non-negative margin, we maximize it. While the optimization formulation is almost identical to the re-solving, our gadget construction is different.

Gadget Game

One way to find the refined strategy is to solve the corresponding linear program. However, algorithms that are tailor-made for extensive form games often outperform the optimization approach (Bošanskỳ 2013). These algorithms often permit the use of domain-specific tricks to provide further performance gains (Johanson et al. 2012). Thus, formulating our optimization problem $LP3$ as an extensive form game will mean that we can compute larger subgame abstractions using the available computing resources. Essentially, the construction of a Gadget 3 corresponding to the $LP3$ will allow us to compute larger subgames than would be possible if we simply used $LP3$. We now provide the construction of such a gadget game.

Gadget Game Construction

All states in the original subgame are directly copied into the resulting gadget game. We create the gadget game by making two alterations to the original subgame. (i) we shift p_2's utilities using the CBV_2 (To initialize all p_2 values to zero) and (ii) we add a p_2 node followed by chance nodes at the top of the subgame (to allow the opponent to pick any starting state, relating the game values to margin) We will distinguish the states, strategies, utilities, etc. for the gadget game by adding a tilde to corresponding notation. The following is a description of the steps (see also Figure 4 that visualizes the constructed Gadget 3)

1. We establish a common baseline. To compare the changes in the performance of each of p_2's root information sets, it is necessary to give them a common baseline. We use the original strategy σ_1^S as the starting point. For every $I \in \mathcal{I}_2^{R(S)}$, we subtract the opponent's original counterfactual best response value, setting the utility at each terminal node $z \in Z(I)$ to $\tilde{u_2}(z) = u_2(z) - CBV_2^{\sigma_1^S}(I)$ (we also update $\tilde{u_1}(\tilde{z}) = -\tilde{u_2}(\tilde{z})$ since we need the game to remain zero-sum). This shifting gives all of our opponent's starting states a value of zero if we do not deviate from our original strategy σ_1^S .
2. p_2 is permitted to choose his belief at the start of the subgame, while p_1 retains his belief from the original strategy

at the point where the subgame begins. Since p_2 is aiming to maximize $\tilde{u}_2$, he will always select the information set with the lowest margin. The minimax nature of the zero-sum game forces p_1 to find a strategy that maximizes this value. We add additional decision node $\tilde{d}$ for p_2. Each action corresponds to choosing an information set I to start with, but we do not connect this action directly to this state. Instead, each action leads to a new chance node $s_{\tilde{I}}$, where the chance player chooses the histories $h \in \tilde{I}$ based on the probability $\pi^{\sigma}_{-2}(h)$.

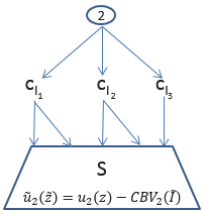

Figure 4: Max margin gadget - Gadget 3. Notice that given the original strategy of p_1 , opponent's best response utility is zero (thanks to the offset of terminal utilities).

Lemma 2. *Strategy for the Gadget* 3 *is Nash Equilibrium if and only if it's a solution to the LP3*

Follows from the construction of the Gadget 3*.*

Experiments

In this section, we evaluate endgame solving, re-solving and max-margin subgame refinement on the safe-refinement task for a large-scale game. We use an improved version of the Nyx agent, the second strongest participant at the 2014 Annual Computer Poker Competition (heads-up no-limit Texas Hold'em Total Bankroll) as the baseline strategy to be refined in subgames.

All three of the subgame refinement techniques tested here used the same abstractions and trunk strategy. Following (Ganzfried and Sandholm 2015), we begin the subgame at start of the last round (the river). While we used card abstraction to compute the original (trunk) strategy (specifically (Schmid et al. 2015) and (Johanson et al. 2013)), the fine-grained abstraction for the endgame is calculated without the need for card abstraction. This is an improvement over the original implementation (Ganzfried and Sandholm 2015), where both the trunk strategy and the refined subgame used card abstraction. This is a result of the improved efficiency of the CFR+ algorithm (and the domain-specific speedups it enables), whereas the endgame solving in (Ganzfried and Sandholm 2015) used linear programming to compute the strategy.

The original strategy uses action abstraction with up to 16 actions in an information set. While this number is relatively large compared to other participating agents, it is still distinctly smaller compared to the best-known upper-bound on the size of the support of an optimal strategy (Schmid, Moravcik, and Hladik 2014). In contrast to the action abstraction used for the original Nyx strategy that uses imperfect recall for the action abstraction, the refined subgame uses perfect recall. We use the same actions in the refined subgame as in the original strategy.

We refine only the subgames that (after creating the fine-grained abstraction) are smaller than $1,000$ betting sequences - this is simply to speed up the experiments. The original agent strategy is used for both p_1 and p_2 in the trunk of the game. Once gameplay reaches the subgame (river), we refine the $P1$ strategy using each of the three techniques. We ran $10,000$ iterations of the CFR+ algorithm in the corresponding gadget games. Exponential weighting is used to update the average strategies (Tammelin et al. 2015). Each technique was used to refine around $2,000$ subgames. Figure 5 visualizes the average margins for the evaluated techniques.

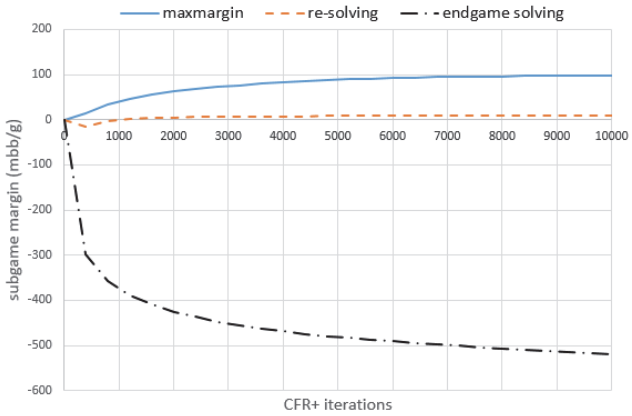

Figure 5: Subgame margins of the refined strategies. One big blind corresponds to 100 chips. The max-margin technique produces the optimal value. We see that the optimal value is much greater than the one produced by either re-solving or endgame solving (which produces even negative margins). The 95% confidence intervals for the results (after $10,000$ iterations) are: maxmargin 101.49 ± 7.09, re-solving 8.79 ± 2.45, endgame solving -518.5 ± 49.19

Endgame Solving The largely negative margin values for the endgame solving suggest that the produced strategy may indeed be much more exploitable.

Re-solving The positive margin for re-solving shows that, although there's no explicit construction that forces the margin to be greater than zero, it does increase in practice. Notice, however, that the margin is far below the optimal level.

Max-margin Refinement This technique produces a much larger subgame margin than the previous techniques. The size of the margin suggests that the original strategy is potentially quite exploitable, and our technique can substantially decrease the exploitability - see Theorem 1.

Conclusion

We have introduced max-margin subgame refinement, a new technique for subgame refinement of large imperfect information games. The subgame margin is a well-motivated value with appealing properties for endgame solving, namely regarding the resulting exploitability. We for-

malized and proved these properties in Theorem 1. As the name of the our technique suggests, the technique aims to maximize this well-motivated value. We also formulated our approach using both linear optimization and extensive form game (gadget) construction. Experimental results have confirmed that our gadget game successfully finds refined strategies with substantially larger margins than previous approaches. The rather large values of the margin that the technique provided suggest that even though we evaluated the technique using a state-of-the-art strategy, such strategies still contain tremendous space for improvement in such large games.

Acknowledgments

The work was supported by the Czech Science Foundation Grant P402/13-10660S and by the Charles University (GAUK) Grant no. 391715. Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

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 *International Joint Conference on Artificial Intelligence*, 661–668.

Bošanskỳ, B. 2013. Solving extensive-form games with double-oracle methods. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems*, 1423–1424.

Bowling, M.; Burch, N.; Johanson, M.; and Tammelin, O. 2015. Heads-up limit holdem poker is solved. *Science* 347(6218):145–149.

Burch, N.; Johanson, M.; and Bowling, M. 2013. Solving imperfect information games using decomposition. *arXiv preprint arXiv:1303.4441*.

Čermák, J.; Bošanskỳ, B.; and Lisy, V. 2014. Practical performance of refinements of nash equilibria in extensive-form zero-sum games. In *Proceedings of the European Conference on Artificial Intelligence*.

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*.

Ganzfried, S., and Sandholm, T. 2015. Endgame solving in large imperfect-information games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 37–45.

Gibson, R. 2014. Regret minimization in games and the development of champion multiplayer computer poker-playing agents. *Ph.D. Dissertation, University of Alberta*.

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*, volume 21, 1007.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2007. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, 50. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press.

Johanson, M.; Bard, N.; Lanctot, M.; Gibson, R.; and Bowling, M. 2012. Efficient nash equilibrium approximation through monte carlo counterfactual regret minimization. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 837–846.

Johanson, M.; Burch, N.; Valenzano, R.; and Bowling, M. 2013. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, 271–278.

Johanson, M. 2013. Measuring the size of large no-limit poker games. *arXiv preprint arXiv:1302.7008*.

Müller, M., and Gasser, R. 1996. Experiments in computer Go endgames. *Games of No Chance* 273–284.

Müller, M. 2002. Computer Go. *Artificial Intelligence* 134(1):145–179.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic Game Theory*, volume 1. Cambridge University Press Cambridge.

Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. MIT press.

Pita, J.; Jain, M.; Ordónez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2009. Using game theory for Los Angeles airport security. *AI Magazine* 30(1):43.

Sandholm, T. 2010. The state of solving large incomplete-information games, and application to poker. *AI Magazine* 31(4):13–32.

Schmid, M.; Moravcik, M.; Hladik, M.; and Gaukroder, S. J. 2015. Automatic public state space abstraction in imperfect information games. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Schmid, M.; Moravcik, M.; and Hladik, M. 2014. Bounding the support size in extensive form games with imperfect information. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Tammelin, O.; Burch, N.; Johanson, M.; and Bowling, M. 2015. Solving heads-up limit Texas holdem. *Technical report, University of Alberta*.

Waugh, K.; Schnizlein, D.; Bowling, M.; and Szafron, D. 2009. Abstraction pathologies in extensive games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 781–788.