Limited Lookahead in Imperfect-Information Games

Christian Kroer
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, Pa, USA
ckroer@cs.cmu.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, Pa, USA
sandholm@cs.cmu.edu

ABSTRACT

Limited lookahead has been studied for decades in complete-information games. We initiate a new direction via two simultaneous deviation points: generalization to incomplete-information games and a game-theoretic approach. We study how one should act when facing an opponent whose lookahead is limited. We study this for opponents that differ based on their lookahead depth, based on whether they, too, have incomplete information, and based on how they break ties. We characterize the hardness of finding a Nash equilibrium or an optimal commitment strategy for either player, showing that in some of these variations the problem can be solved in polynomial time while in others it is PPAD-hard or NP-hard. We proceed to design algorithms for computing optimal commitment strategies—for when the opponent breaks ties favorably, according to a fixed rule, or adversarially. We then experimentally investigate the impact of limited lookahead. The limited-lookahead player often obtains the value of the game if she knows the expected values of nodes in the game tree for some equilibrium—but we prove this is not sufficient in general. Finally, we study the impact of noise in those estimates and different lookahead depths. This uncovers an incomplete-information game lookahead pathology.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems;
J.4.a [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Theory, Economics

Keywords

Game theory, equilibrium finding, limited lookahead

1. INTRODUCTION

Limited lookahead has been a central topic in AI game playing for decades. To date, it has been studied in single-agent settings and complete-information games—specifically in well-known games such as chess, checkers, Go, etc., as well as in random game tree models [11, 22, 23, 20, 21, 2, 24, 25]. In this paper, we initiate the game-theoretic study of limited lookahead in incomplete-information games. Such games are significantly more broadly applicable to practical settings—for example auctions, negotiations, military settings, security, cybersecurity, and medical settings—than complete-information games. Mirrokni, Thain, and Vetta [19] conducted a game-theoretic analysis of lookahead, but they consider only complete-information games, and the results are for four specific games rather than broad classes of games. Instead, we analyze the questions for incomplete information and for general

games. Specifically, we study general-sum extensive-form games. As is typical in the literature on limited lookahead in complete-information games, we derive our results for a two-agent setting. One agent is a rational player (Player r) trying to optimally exploit a limited-lookahead player (Player l).

Our results extend immediately to one rational player and more than one limited-lookahead player, as long as the latter all break ties according to the same scheme (statically, favorably, or adversarially—as described later in the paper). This is because such a group of limited-lookahead players can be treated as one from the perspective of our results.

The type of limited-lookahead player we introduce is quite natural and analogous to that in the literature on complete-information games. Specifically, we let the limited-lookahead player l have a node evaluation function that places numerical values on all nodes in the game tree. Given a strategy for the rational player, at each information set at some depth i, Player l picks an action that maximizes the expected value of the evaluation function at depth $i + k$, assuming optimal play between those levels.

Our study is the game-theoretic, incomplete-information generalization of lookahead questions studied in the literature and therefore interesting in its own right. The model also has applications such as biological games, where the goal is to steer an evolution or adaptation process (which typically acts myopically with lookahead 1) [28] and security games where opponents are often assumed to be myopic (as makes sense when the number of adversaries is large [32]). Furthermore, investigating how well a rational player can exploit a limited-lookahead player lends insight into the limitations of using limited-lookahead algorithms in multiagent decision making.

We consider the problem of exploiting a limited-lookahead opponent under various assumptions about the opponent, mapping out the hardness of the problem under all these alternatives assumptions. We consider three dimensions: whether the opponent has information sets, whether the opponent has lookahead 1 or more, and whether the opponent breaks ties statically, adversarially, or favorably. If Player l has no information sets, lookahead 1, and breaks ties either adversarially or by a static scheme, we show that both a Nash equilibrium and an optimal strategy to commit to (i.e., a Stackelberg strategy) can be found in polynomial time. Conversely, if any of these assumptions do not hold, we show that equilibrium finding is PPAD-hard and finding an optimal strategy to commit to is NP-hard.

We then design algorithms for finding an optimal strategy to commit to for the unlimited, rational player r. We focus on this rather than equilibrium computation because the latter seems nonsensical in this setting: the limited-lookahead player determining a Nash equilibrium strategy would require her to reason about the whole game for the rational player's strategy, which rings contrary to the limited-lookahead assumption. Furthermore, optimal strate-

gies to commit to are desirable for applications such as biological games (because evolution is responding to what we as the "steerer" are doing) and security games (where the defender typically gets to commit to a strategy). Computing optimal strategies to commit to in standard rational settings has previously been studied in normal-form games [4] and extensive-form games [17], the latter implying some complexity results for our setting as we will discuss.

For the case where the limited-lookahead player breaks ties in favor of Player r, or by some static scheme, we develop a mixed-integer program (MIP) that is a natural extension of the sequence-form linear program (LP) from the two-player zero-sum setting.

Then, we derive an algorithm for solving the setting where the limited-lookahead player breaks ties adversarially. For a *given* set of actions that are optimal for the limited-lookahead player, this ends up being a zero-sum game between the rational player and the tie-breaking rule. We then show how to embed this LP in a MIP that branches on which action set to make optimal for the limited-lookahead player.

We experimentally evaluate the usefulness of exploiting limited-lookahead opponents in two recreational games using our new algorithms. The limited-lookahead player often obtains the value of the game if she knows the expected values of nodes in the game tree for some equilibrium—but we provide a counterexample that shows that this is not sufficient in general. We go on to study the impact of noise in those estimates, and different lookahead depths. We uncover an incomplete-information game lookahead pathology, and show how it can be embedded into any game.

As in the literature on lookahead in complete-information games, a potential weakness of our approach is that we require knowing the h function (but make no other assumptions about what information h encodes). In practice, this function may not be known. As in the perfection-information setting, this can lead to the rational exploiter being exploited. However, many practical settings do not have this problem. For example, biological design games [28] and fare-inspection games [32] involve myopic agents that would not be expected to design strategies that exploit the rational player's errors in beliefs about h. If there are multiple limited-lookahead players, it seems even less likely that they could exploit the rational player in this way, as it may require coordination/cooperation.

In general, this paper can be taken as a prescriptive theory of how one should play against a limited-lookahead player, and how a limited-lookahead player should play, or as an investigation of how badly a best-responding limited-lookahead player can be exploited.

2. EXTENSIVE-FORM GAMES

We start by defining the class of games that the players will play, without reference to limited lookahead. The class is general and standard.

An *extensive-form game* Γ is a tuple $\langle N, A, S, Z, \mathcal{H}, \sigma_0, u, \mathcal{I} \rangle$. N is the set of players. A is the set of all actions in the game. S is a set of nodes corresponding to sequences of actions. They describe a tree with root node $s^r \in S$. At each node s, it is the turn of some Player i to move. Player i chooses among actions A_s, and each branch at s denotes a different choice in A_s. Let t_a^s be the node transitioned to by taking action $a \in A_s$ at node s. The set of all nodes where Player i is active is called S_i. $Z \subset S$ is the set of leaf nodes, where $u_i(z)$ is the utility to Player i of node z. We assume, without loss of generality, that all utilities are non-negative. Z_s is the subset of leaf nodes reachable from a node s. $\mathcal{H}_i \subseteq \mathcal{H}$ is the set of heights in the game tree where Player i acts. $\mathcal{H}_0$ is the set of heights where Nature acts. σ_0 specifies the probability distribution for Nature, with $\sigma_0(s, a)$ denoting the probability of Nature choosing outcome a at node s.

Incomplete information is represented in the game model using information sets. $\mathcal{I}_i \subseteq \mathcal{I}$ is the set of information sets where Player i acts. $\mathcal{I}_i$ partitions S_i. For nodes $s_1, s_2 \in I, I \in \mathcal{I}_i$, Player i cannot distinguish among them, and $A_{s_1} = A_{s_2}$.

We denote by σ_i a *behavioral strategy* for Player i. For each information set $I \in \mathcal{I}_i$, it assigns a probability distribution over A_I, the actions at the information set. $\sigma_i(I, a)$ is the probability of playing action a. A *strategy profile* $\sigma = (\sigma_0, \ldots, \sigma_n)$ consists of a behavioral strategy for each player. We will often use $\sigma(I, a)$ to mean $\sigma_i(I, a)$, since the information set specifies which Player i is active. As described above, randomness external to the players is captured by the Nature outcomes σ_0. Using this notation allows us to treat Nature as a player when convenient, although Nature selects actions according to fixed probabilities.

Let the probability of going from node s to node $\hat{s}$ under strategy profile σ be $\pi^\sigma(s, \hat{s}) = \Pi_{\langle \bar{s}, \bar{a} \rangle \in X_{s, \hat{s}}} \sigma(\bar{s}, \bar{a})$ where $X(s, \hat{s})$ is the set of pairs of nodes and actions on the path from s to $\hat{s}$. We let the probability of reaching node s be $\pi^\sigma(s) = \pi^\sigma(s^r, s)$, the probability of going from the root node to s. Let $\pi^\sigma(I) = \sum_{s \in I} \pi^\sigma(s)$ be the probability of reaching any node in I. $\pi_i^\sigma(I) = \pi_i^\sigma(s) \forall s \in I$ due to perfect recall. For probabilities over Nature, $\pi_0^\sigma = \pi_0^{\bar{\sigma}}$ for all $\sigma, \bar{\sigma}$, so we can ignore the strategy profile superscript and write π_0. Finally, for all behavioral strategies, the subscript $-i$ refers to the same definition, excluding Player i. For example, $\pi_{-i}^\sigma(s)$ denotes the probability of reaching s over the actions of the players other than i, that is, if i played to reach s with probability 1.

3. MODEL OF LIMITED LOOKAHEAD

We now describe our model of limited lookahead, which we consider to be very intuitive.

We use the term optimal *hypothetical* play to refer to the way the limited-lookahead agent thinks she will play when looking ahead from a given information set. In actual play part way down that plan, she may change her mind because she will then be able to see to a deeper level of the game tree (given that her lookahead depth is still the same and she will be at a deeper part of the tree).

Let k be the lookahead of Player l, and $S_{I,a}^k$ the nodes at the lookahead depth k below information set I that are reachable (through some path) by action a. As in prior work in the complete-information game setting, Player l has a node-evaluation function $h : S \to \mathbb{R}$ that assigns a heuristic numerical value to each node in the game tree.

Given a strategy σ_r for the other player and fixed action probabilities for Nature, Player l chooses, at any given information set $I \in \mathcal{I}_l$ at depth i, a (possibly mixed) strategy whose support is contained in the set of actions that maximize the expected value of the heuristic function at depth $i + k$, assuming optimal hypothetical play by her ($\max_{\sigma_l}$ in the formula below). We will denote this set by $A_I^* =$

$$\{a : a \in \arg\max_{a \in A_I} \max_{\sigma_l} \sum_{s \in I} \frac{\pi^{\sigma_{-l}}(s)}{\pi^{\sigma_{-l}}(I)} \sum_{s' \in S_{I,a}^k} \pi^\sigma(t_a^s, s') h(s')\},$$

where $\sigma = \{\sigma_l, \sigma_r\}$. Here moves by Nature are also counted toward the depth of the lookahead of the limited-lookahead player, and when looking through such nodes, that player takes an expectation over Nature's moves at that node.

The model is flexible as to how the rational player chooses σ_r and how the limited-lookahead player chooses a (possibly mixed) strategy with supports within the sets A_I^*. For one, we can have these choices be made for both players simultaneously according to the Nash equilibrium solution concept, so neither player wants to change her choices given that the other does not change. As another example, we can ask how the players should make those choices if one of the players gets to make, and commit to, all her choices before the other. This begets multiple settings based on which player

gets to commit first and how ties are broken. We will study all of the above variants. Other solution concepts and refinements could also be used.

4. COMPLEXITY

In this section we analyze the complexity of finding strategies according to these solution concepts.

4.1 Nash equilibrium

Finding a Nash equilibrium when Player l either has information sets containing more than one node, or has lookahead at least 2, is PPAD-hard. This is because finding a Nash equilibrium in a 2-player general-sum normal-form game is PPAD-hard [3], and any such game can be converted to a depth 2 extensive-form game (where the second player does not know what the first player moved), where the general-sum payoffs are the evaluation function values.

If the limited-lookahead player only has singleton information sets and lookahead 1, an optimal strategy can be trivially computed in polynomial time in the size of the game tree for the limited-lookahead player (without even knowing the other player's strategy σ_r) because for each of her information sets, we simply have to pick an action that has highest immediate heuristic value. To get a Nash equilibrium, what remains to be done is to compute a best response for the rational player, which can also be easily done in polynomial time [8].

4.2 Commitment strategies

Next we study the complexity of finding commitment strategies. The complexity depends on whether the game has incomplete information (information sets that include more than one node) for the limited-lookahead player, how far that player can look ahead, and how she breaks ties in her action selection.

No information sets, lookahead 1, static tie-breaking As for the Nash equilibrium case, if the limited-lookahead player only has singleton information sets and lookahead 1, an optimal strategy can be trivially computed in polynomial time. We can use the same approach, except that the specific choice among the actions with highest immediate value is dictated by the tie-breaking rule. With this strategy in hand, finding a utility-maximizing strategy for Player r again consists of computing a best response.

No information sets, lookahead 1, adversarial tie-breaking When ties are broken adversarially, the choice of response depends on the choice of strategy for the rational player. The set of optimal actions A_s^* for any node $s \in S_l$ can be precomputed, since Player r does not affect which actions are optimal. Player l will then choose actions from these sets to minimize the utility of Player r. We can view the restriction to a subset of actions as a new game, where Player l is a rational player in a zero-sum game. An optimal strategy for Player r to commit to is then a Nash equilibrium in this smaller game. This is solvable in polynomial time by an LP that is linear in the size of the game tree [30], and algorithms have been developed for scaling to large games [7, 33, 15, 12, 13, 6, 5].

No information sets, lookahead 1, favorable tie-breaking In this case, Player l picks the action from A_s^* that maximizes the utility of Player r. Perhaps surprisingly, computing the optimal solution in this case is harder than when facing an adversarial opponent. All our hardness proofs, presented in the appendix, are by reduction from 3SAT.

DEFINITION 1. *A 3SAT instance consists of a tuple (V, C). V is a set of n Boolean variables, and C is a set of m clauses of the form $(l_1 \vee l_2 \vee l_3)$ where each l_i represents a literal requiring some variable to be true or false.*

THEOREM 1. *Computing a utility-maximizing strategy for the rational player to commit to is NP-hard if the limited-lookahead player breaks ties in favor of the rational player.*

PROOF. We reduce from 3SAT. A picture illustrating our reduction is given in Figure 1, and a description is given below.

Let the root node be a chance node. It chooses with equal probability between $|C|$ child nodes, each representing a clause. Each such descendant clause node is a singleton information set belonging to Player l. Each clause node has three actions, representing the three literals in the clause. Each such action leads to a node representing that literal. Player l gets the same value from each action and is therefore indifferent. Player r acts at each literal node, with all literal nodes representing the same variable being in an information set together. Thus, Player r has an information set for each variable. At each variable information set, there is a true and false action. For a given literal node in some variable information set, the true action leads a payoff of 1 if the literal requires the variable to be true, and 0 otherwise. Similarly, the false action leads to a payoff of 1 if the literal requires the variable to be false, and 0 otherwise.

The decision problem is then: does there exist a strategy for Player r with expected payoff 1? This is the case if and only if the strategy for Player r represents a satisfying assignment to V, C, as each clause must have some action available where a satisfying assignment for the literal is chosen with probability 1. □
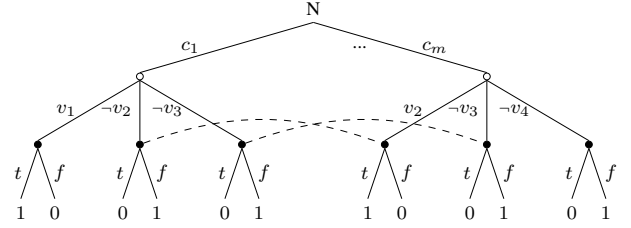
Figure 1: The game tree in our proof of Theorem 1. Dashed lines denote information sets.

No information sets, lookahead > 1, favorable tie-breaking It is NP-hard to compute an optimal strategy to commit to in extensive-form games when both players are rational [17]. That was proven by reducing from knapsack to a 2-player perfect-information game of depth 4. This immediately gives us two results: (1) finding an optimal strategy for Player r to commit to is NP-hard if Player l has lookahead at least 4, and (2) computing an optimal strategy to commit to for Player l is NP-hard even with lookahead 1. Their result also implies NP-hardness of computing a strategy to commit to for the rational player, if our l player has lookahead of at least 4. We tighten this to lookahead 2:

THEOREM 2. *Computing a utility-maximizing strategy for the rational player to commit to is NP-hard if the limited lookahead player has lookahead at least 2.*

PROOF. We reduce from 3SAT. We use the same reduction as for Theorem 1, except that at each clause node, we also add an "unsatisfied" action that leads directly to a leaf node with payoff 0 for Player r and payoff $\frac{2}{3}$ for Player l.

For all leaf nodes under a variable node, we set the payoff to 1 for Player r, and 0 or 1 for Player l, for when the leaf represents the ancestor literal being unsatisfied or satisfied, respectively. The modifications are shown for a single clause in Figure 2.

The question is whether Player r can compute a strategy such that Player l selects a literal action for each clause, assuming that Player l breaks ties such that the unsatisfied action is least preferred. For a given variable, choosing a strategy strictly between $0, \frac{2}{3}$ for

the two actions leads to zero utility gain, since Player 2 will then always prefer the unsatisfied actions over any literal belonging to the variable. Thus we can assume that Player r plays a pure strategy, since at most one action can have its probability set high enough to yield utility gain. Now, for each clause, Player l will only choose a literal action if that variable is set to the correct value to satisfy the clause. Thus, if Player r can compute a strategy that gives expected utility 1, each clause node must have at least one variable with a satisfying assignment. □
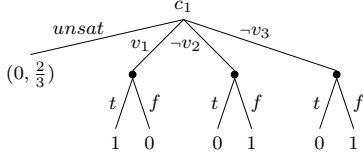

Figure 2: The clause modification in our proof of Theorem 2.

Limited-lookahead player has information sets When the limited lookahead player has information sets, we show that computing a strategy to commit to is NP-hard:

THEOREM 3. *Computing a utility-maximizing strategy for the rational player to commit to is NP-hard if the limited lookahead player has information sets of at least size* 6.

PROOF. We reduce from 3SAT. Let the root node be a chance node. It chooses with equal probability between all variable and clause pairs v, c such that $v \in c$. Player r acts at each child node, being able to distinguish only which variable was chosen. For each information set, Player 1 can choose between a true and a false action, representing setting the associated variable to true or false, respectively. At the next level where Player l is active. The information sets at the level are constructed as follows. For each $c \in C$ an information set is constructed, containing all nodes representing Player r choosing both true and false for each $v \in c$. For each information set representing some clause c, Player l has 4 actions. First is an unsat action, leading to payoff 0 for Player 1 and payoff 1 for Player l, no matter which node in the information set play has reached. Second, an action for each variable $v \in c$ leading to payoff 1 for Player r, and payoff 3 to Player l if play reached a node representing v with true or false chosen such that it satisfies c, and payoff 0 for all other nodes in the information set.

We claim that there is a satisfying assignment if and only if Player 1 can commit to a strategy with expected payoff 1. Let $\phi : V \to \{true, false\}$ be a satisfying assignment to V, C. Let Player r deterministically pick actions at each variable information set according to ϕ. If play reaches a singleton node, Player l has only one action available, guaranteeing payoff 1. If play reaches some information set representing a clause c, Player l has expected payoff of $3 \cdot \frac{1}{3}$ when picking any action representing a satisfied literal $l \in c$, as the conditional probability of being at a node representing $v(l)$ is $\frac{1}{3}$, and Player r chooses the satisfying action with probability 1. Since Player l breaks ties such that unsatisfied actions are least preferred, she will pick an action representing a variable for each information set, yielding payoff 1 to Player r. This covers all possible outcomes, giving an expected payoff of 1 to Player r.

Given some strategy for Player r that gives payoff 1 in expectation, we show how to construct a satisfying assignment to V, C. For a strategy to have payoff 1, Player l must be choosing variable actions at each information set for some clause c. This is the case if and only if Player r selects the satisfying truth value with probability 1 for some $v \in c$, since the expected payoff of taking a variable action is otherwise strictly smaller than the unsatisfied action. This leads directly to a satisfying assignment, by choosing the corresponding value assignment for each action that is selected with

probability 1, and choosing an arbitrary assignment for every other variable. □
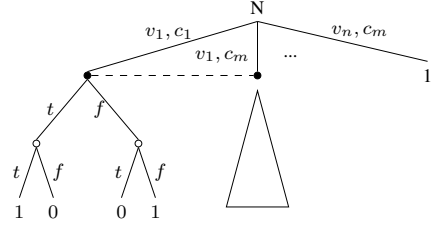

Figure 3: The game tree for our proof of Theorem 3.

5. ALGORITHMS

We showed how to compute an optimal strategy to commit to in polynomial time when the limited-lookahead player has no information sets, lookahead 1, and ties are broken either by a static scheme or adversarially. We then showed hardness for all other cases. In this section we will develop worst-case exponential-time algorithms for solving the hard commitment-strategy cases. Here we focus on commitment strategies rather than the hard Nash equilibrium problem classes because Player l playing a Nash equilibrium strategy would require the limited-lookahead player l to reason about the whole game for the opponent's strategy, which rings contrary to the limited-lookahead assumption. Further, optimal strategies to commit to are desirable for applications such as biological games (because evolution is responding to what we as the "steerer" are doing) and security games (where the defender typically gets to commit to a strategy).

Favorable tie breaking

We start with the case where the limited-lookahead player breaks ties in the rational player's favor. We use the idea of the sequence form [26, 9, 10], where a variable is introduced for each *sequence* (information set-action pair) of actions a given player can take. The insight is that in perfect-recall games, a given action at some information set for Player i is reached by a unique sequence of actions of Player i. This is exploited to represent the probability $\pi_i^\sigma(I)\sigma(I, a)$ of a given action $a \in A_I$ being realized by a variable x_a. To ensure that a valid set of realization probabilities is computed, the constraint $x_a = \sum_{a' \in A_{I'}} x_{a'}$ is introduced for all information sets I' and actions a such that a is the last action by Player i on the path to I'. A behavioral strategy is then obtained simply by normalizing by the realization probability of the last action a: $\sigma(I', a') = \frac{x_{a'}}{x_a}$. With this formulation, duality is used to obtain a linear program for computing Nash equilibria in zero-sum extensive-form games.

In our case, we cannot apply duality. Instead, we work directly on the sequence form variables for both players. For Player r, we introduce realization variables $x_a \in [0, 1]$ for each action a. For Player l, we introduce Boolean realization variables $y_a \in \{0, 1\}$ for each action a, as there always exists a pure strategy that maximizes utility, given a strategy for the other player. This is a key deviation from the traditional sequence form, where the variables are real valued.

For any node s, we have $\pi_1(s) = x_a, \pi_2(s) = y_{a'}$ where actions a, a' are the last actions on the path to s for Player r and Player l, respectively. Using this notation, we introduce a variable r_z representing the expected utility from each leaf node z. The expected utility of a leaf node requires computing the probability of it being reached $\pi_0(z) \cdot \pi_1(z) \cdot \pi_2(z)$, which is a non-linear function. However, since Player l uses only probabilities 0 and 1, we can separate this into two linear single-variable constraints

$$r_z \le u_1(z)\pi_0(z)\pi_1(z) \quad \text{and} \quad r_z \le u_1(z)\pi_2(z)$$

The objective function is then simply $\sum_{z \in Z} r_z$.

Finally, we must ensure that the strategy chosen for Player l maximizes her utility according to the evaluation function at each information set $I \in \mathcal{I}_2$. Let $S_{I,a}^k$ be the set of nodes at depth k below I, reachable from information set I when taking action a. Letting $\pi^\sigma(s)$ denote the probability of reaching s under optimal hypothetical play, we introduce the following constraint for all $a, a' \in A_I$:

$$\sum_{s \in S_{I,a}^k} \pi^\sigma(s) h(s) \geq \sum_{s \in S_{I,a'}^k} \pi^\sigma(s) h(s) - M(1 - y_a) \quad (1)$$

The constraint requires that the weighted sum over descendant node evaluation function values is at least as high at a as at any other action a'. The negative term ensures that the constraint is active only if the action is chosen ($y_a = 1$) by subtracting a sufficiently large number M otherwise.

The number of MIP matrix entries needed to implement this sparsely is $O(\sum_{I \in \mathcal{I}_l} |A_I| \cdot \max_{s \in S} |A_s|^{\min\{k, k'\}})$, where k' is the maximum depth of the subtrees rooted in I. We present the details on the implementation and the proof of the MIP size in the proof of the similar case for Theorem 4. For many games, the lookahead depth k, maximum action set size, and number of information sets would all be much smaller than the size of the game tree $|S|$. For example, in the largest game that we investigate in the experimental section, the above expression, which is an upper bound, yields 448 entries. The game tree has 199 nodes. The MIP is thus almost linear in the size of the game tree for many realistic games.

Static tie-breaking When the limited-lookahead player breaks ties according to some static scheme $\succ$, we can use the same approach as for favorable tie breaking, except that Equation 1 has to be a strict inequality for any a, a' such that $a' \succ a$. This can be achieved in a MIP by subtracting sufficiently small ϵ.

Adversarial tie breaking When the limited-lookahead player breaks ties adversarially, we wish to compute a strategy that maximizes the worst-case best response by the limited-lookahead player.

For arguments sake, say that we were given $\mathcal{A}$, which is a fixed set of pairs, one for each information set I of the limited-lookahead player, consisting of a set of optimal actions A_I^* and one strategy for hypothetical play σ_l^I at I. Formally, $\mathcal{A} = \bigcup_{I \in \mathcal{I}_l} \langle A_I^*, \sigma_l^I \rangle$. To make these actions optimal for Player l, Player r must choose a strategy such that all actions in $\mathcal{A}$ are best responses according to the evaluation function of Player l. Formally, for all $a, a^* \in \mathcal{A}, a' \notin \mathcal{A}$ (letting $\pi(s)$ denote probabilities induced by σ_l^I for the hypothetical play between I, a and s):

$$\sum_{s \in S_{I,a}^k} \pi(s) \cdot h(s) > \sum_{s \in S_{I,a'}^k} \pi(s) \cdot h(s) \quad (2)$$

$$\sum_{s \in S_{I,a}^k} \pi(s) \cdot h(s) = \sum_{s \in S_{I,a^*}^k} \pi(s) \cdot h(s) \quad (3)$$

Player r has to choose a worst-case utility-maximizing strategy that satisfies Equations 2 and 3, and Player l has to compute a (possibly mixed) strategy from $\mathcal{A}$ such that the utility of Player r is minimized. We show that this problem can be solved by LP 8.

THEOREM 4. *For some fixed choice of actions $\mathcal{A}$ to force Player l to play, Nash equilibria of the induced game can be computed in polynomial time by a linear program that has size $O(|S|) + O(\sum_{I \in \mathcal{I}_l} |A_I| \cdot \max_{s \in S} |A_s|^{\min\{k, k'\}})$.*

To prove this theorem, we first design a series of linear programs for computing best responses for the two players. We will then use duality to prove the theorem statement.

In the following, it will be convenient to change to matrix-vector

notation. Our notation will be analogous to that of von Stengel [30], with some extensions. Let $A = -B$ be matrices describing the utility function for Player r and the maximization problem of Player l over $\mathcal{A}$, respectively. Rows are indexed by Player r sequences, and columns by Player l sequences. For sequence form vectors x, y, the objectives to be maximized for the players are then xAy, xBy. Matrices E, F are used to describe the sequence form constraints for Player r and l, respectively. Rows correspond to information sets, and columns correspond to sequences. Letting e, f be standard unit vectors of length $|\mathcal{I}_r|, |\mathcal{I}_l|$, respectively, the constraints $Ex = e, Fy = f$ describe the sequence form constraint for the respective players. Given a strategy x for Player r satisfying Equations 2 and 3 for some $\mathcal{A}$, the optimization problem for Player l becomes choosing a vector of y' representing probabilities for all sequences in $\mathcal{A}$ that minimize the utility of Player r. Letting a prime superscript denote the restriction of each matrix and vector to sequences in $\mathcal{A}$, this gives the following primal (4) and dual (5) LPs:

$$\max_{y'} \quad (x^T B') y' \qquad\qquad \min_{q'} \quad q'^T f'$$
$$F' y' = f' \qquad (4) \qquad\qquad q'^T F' \geq x^T B' \qquad (5)$$
$$y \geq 0$$

Where q' is a vector with $|\mathcal{A}| + 1$ dual variables. Given some strategy y' for Player l, Player r maximizes utility among strategies that induce $\mathcal{A}$. This gives the following best-response LP for Player r:

$$\max_{x} \quad x^T (Ay')$$
$$x^T E^T = e^T$$
$$x \geq 0 \qquad\qquad\qquad (6)$$
$$x^T H_{\neg \mathcal{A}} - x^T H_{\mathcal{A}} \leq -\epsilon$$
$$x^T G_{\mathcal{A}^*} = x^T G_{\mathcal{A}}$$

Where the last two constraints encode equations 2 and 3, respectively. The dual problem uses the unconstrained vectors p, v and constrained vector u and looks as follows

$$\min_{p,u,v} \quad e^T p - \epsilon \cdot u$$
$$E^T p + (H_{\neg \mathcal{A}} - H_{\mathcal{A}}) u + (G_{\mathcal{A}^*} - G_{\mathcal{A}}) v \geq A' y' \qquad (7)$$
$$u \geq 0$$

We can now merge the dual (5) with the constraints from the primal (6) to compute a solution where Player r chooses x, which she will choose to minimize the objective of (5), a minmax strategy:

$$\min_{x,q'} \quad q'^T f'$$
$$q'^T F' - x^T B' \geq 0$$
$$-x^T E^T = -e^T$$
$$x \geq 0 \qquad\qquad\qquad (8)$$
$$x^T H_{\mathcal{A}} - x^T H_{\neg \mathcal{A}} \geq \epsilon$$
$$x^T G_{\mathcal{A}} - x^T G_{\mathcal{A}^*} = 0$$

Taking the dual of this gives

$$\max_{y',p} \quad -e^T p + \epsilon \cdot u$$
$$-E^T p + (H_{\mathcal{A}} - H_{\neg \mathcal{A}}) u + (G_{\mathcal{A}} - G_{\mathcal{A}^*}) v \leq B' y' \qquad (9)$$
$$F' y' = f'$$
$$y, u \geq 0$$

We are now ready to prove Theorem 4.

PROOF. The LPs are (8) and (9). We will use duality to show that they provide optimal solutions to each of the best response LPs. Since $A = -B$, the first constraint in (9) can be multiplied by -1 to obtain the first constraint in (7) and the objective function can be transformed to that of (7) by making it a minimization. By the weak duality theorem, we get the following inequalities

$$q'^T f' \geq x^T B' y'; \text{ by LPs 4 and 5}$$
$$e^T p - \epsilon \cdot u \geq x^T A' y'; \text{ by LPs 6 and 7}$$

We can multiply the last inequality by -1 to get:

$$q'^T f' \geq x^T B' y' = -x^T A' y' \geq -e^T p + \epsilon \cdot u \quad (10)$$

By the strong duality theorem, for optimal solutions to LPs 8 and 9 we have equality in the objective functions $q'^T f' = -e^T p + \epsilon u$ which yields equality for Equation 10, and thereby equality for the objective functions in LPs 4, 5 and for 6, 7. By strong duality, this implies that any primal solution x, q' and dual solution y', p to LPs 8 and 9 yields optimal solutions to the LPs 4 and 6. Both players are thus best responding to the strategy of the other agent, yielding a Nash equilibrium.

Conversely, any Nash equilibrium gives optimal solutions x, y' for LPs 4 and 6. With corresponding dual solutions p, q', equality is achieved in Equation 10, meaning that LPs 8 and 9 are solved optimally.

It remains to show the size bound for LP 8. Using sparse representation, the number of non-zero entries in the matrices A, B, E, F is linear in the size of the game tree.

The constraint set $x^T H_{\mathcal{A}} - x^T H_{\neg \mathcal{A}} \geq \epsilon$, when naively implemented, is not. The value of a deactivated sequence at some information set I is dependent on the choice among the cartesian product of choices at each information set I' encountered in hypothetical play below it. In practice we can avoid this by having a real-valued variable $v_I^d(I')$ representing the value of I' in lookahead from I, and introducing constraints

$$v_I^d(I') \geq v_I^d(I', a)$$

for each $a \in I'$, where $v_I^d(I', a)$ is a variable representing the value of taking a at I'. If there are more information sets below I' where Player l plays, before the lookahead depth is reached, we recursively constrain $v_I^d(I', a)$ to be:

$$v_I^d(I', a) \geq \sum_{\check{I} \in \mathcal{D}} v_I^d(\check{I}) \quad (11)$$

where $\mathcal{D}$ is the set of information sets at the next level where Player l plays. If there are no more information sets where Player l acts, then we constrain $v_I^d(I', a)$:

$$v_I^d(I', a) \geq \sum_{s \in S_{I', a}^k} \pi_{-l}^\sigma h(s) \quad (12)$$

Setting it to the probability-weighted heuristic value of the nodes reached below it.

Using this, we can now write the constraint that a dominates all $a' \in I, a' \notin \mathcal{A}$ as:

$$\sum_{s \in S_{I, a}^k} \pi^\sigma(s) h(s) \geq v_I^d(I)$$

There can at most be $O(\sum_{I \in \mathcal{I}_l} |A_I|)$ actions to be made dominant. For each action at some information set I, there can be at most $O(\max_{s \in S} |A_s|^{\min\{k, k'\}})$ entries over all the constraints, where k' is the maximum depth of the subtrees rooted at I. This is because each node at the depth the player looks ahead to has its heuristic

value added to at most one expression.

For the constraint set $x^T G_{\mathcal{A}} - x^T G_{\mathcal{A}^*} = 0$, the choice of hypothetical plays has already been made for both expressions, and so we have the constraint

$$\sum_{s \in S_{I, a}^k} \pi^\sigma(s) h(s) = \sum_{s \in S_{I, a'}^k} \pi^{\sigma'}(s) h(s)$$

for all $I \in \mathcal{I}_l, a, a' \in I, \{a, \sigma^l\}, \{a', \sigma^{l,'}\} \in \mathcal{A}$, where

$$\sigma = \{\sigma_{-l}, \sigma^l\}, \sigma' = \{\sigma_{-l}, \sigma^{l,'}\}$$

There can at most be $\sum_{I \in \mathcal{I}_l} |A_I|^2$ such constraints. Which is dominated by the size of the previous constraint set.

Summing up gives the desired bound. □

In reality we are not given $\mathcal{A}$. To find a commitment strategy for Player r, we could loop through all possible structures $\mathcal{A}$, solve LP 8 for each one, and select the one that gives the highest value.

We show that this can be done without such exhaustive enumeration. We introduce a MIP formulation that picks the optimal induced game $\mathcal{A}$. The MIP is given in (13). We introduce Boolean sequence-form variables that denote making sequences suboptimal choices. These variables are then used to deactivate subsets of constraints, so that the MIP branches on formulations of LP 8, i.e., what goes into the structure $\mathcal{A}$. The size of the MIP is of the same order as that of LP 8.

$$\begin{aligned}
\min_{x, q, z} \quad & q^T f \\
& q^T F \geq x^T B - zM \\
& Ex = e \\
& x^T H_{\mathcal{A}} \geq x^T H_{\neg \mathcal{A}} + \epsilon - (1 - z)M \\
& x^T G_{\mathcal{A}} = x^T G_{\mathcal{A}^*} \pm (1 - z)M \\
& \sum_{a \in A_I} z_a \geq z_{a'} \\
& x \geq 0, \quad z \in \{0, 1\}
\end{aligned} \quad (13)$$

The variable vector x contains the sequence form variables for Player r. The vector q is the set of dual variables for Player l. z is a vector of Boolean variables, one for each Player l sequence. Setting $z_a = 1$ denotes making the sequence a an inoptimal choice. The matrix M is a diagonal matrix with sufficiently large constants (e.g. the smallest value in B) such that setting $z_a = 1$ deactivates the corresponding constraint. Similar to the favorable-lookahead case, we introduce sequence form constraints $\sum_{a \in A_I} z_a \geq z_{a'}$ where a' is the parent sequence, to ensure that at least one action is picked when the parent sequence is active. We must also ensure that the incentivization constraints are only active for actions in $\mathcal{A}$:

$$x^T H_{\mathcal{A}} - x^T H_{\neg \mathcal{A}} \geq \epsilon - (1 - z)M \quad (14)$$
$$x^T G_{\mathcal{A}} - x^T G_{\mathcal{A}^*} = 0 \pm (1 - z)M$$

For diagonal matrices M with sufficiently large entries. Equality is implemented with a pair of inequality constraints. The $\pm$ denotes adding or subtracting, respectively, for the two inequalities.

The values of each column constraint in equation 14 is implemented by a series of constraints. We add Boolean variables $\sigma_l^I(I', a')$ for each information set action pair I', a' that is potentially chosen in hypothetical play at I. Using our regular notation, for each a, a' where a is the action to be made dominant, the constraint is implemented by:

$$\sum_{s \in S_{I, a}^k} v^i(s) \geq v_I^d(I), \quad v^i(s) \leq \sigma_l^I(I', a') \cdot M \quad (15)$$

where the latter ensures that $v^i(s)$ is only non-zero if chosen in hypothetical play. We further need the constraint $v^i(s) \leq \pi^{\sigma}_{-l}(s)h(s)$ to ensure that $v^i(s)$, for a node s at the lookahead depth, is at most the heuristic value weighted by the probability of reaching s.

Since we have just modified existing constraints, and added variables and entries corresponding to the number of sequences and information sets, the size of this MIP has size on the order of the size of LP 8.

6. EXPERIMENTS

In this section we experimentally investigate how much utility can be gained by optimally exploiting a limited-lookahead player. We take a conservative approach, and assume that ties are broken adversarially. We conduct experiments on Kuhn poker [14], a canonical testbed for game-theoretic algorithms, and a larger simplified poker game that we call KJ.

Kuhn poker consists of a three-card deck: king, queen, and jack. Each player antes 1. Each player is then dealt one of the three cards, and the third is put aside unseen. A round of betting occurs:

- Player 1 can check or bet 1.
  - If Player 1 checks Player 2 can check or raise 1.
    * If Player 2 checks there is a showdown.
    * If Player 2 raises Player 1 can fold or call.
      · If Player 1 folds Player 2 takes the pot.
      · If Player 1 calls there is a showdown for the pot.
  - If Player 1 raises Player 2 can fold or call.
    * If Player 2 folds Player 1 takes the pot.
    * If Player 2 calls there is a showdown.

In a showdown, the player with the higher card wins the pot.

In KJ, the deck consists of two kings and two jacks. Each player antes 1. A private card is dealt to each, followed by a betting round ($p = 2$), then a public card is dealt, follower by another betting round ($p = 4$). If no player has folded, a showdown occurs. Each round of betting looks as follows:

- Player 1 can check or bet p.
  - If Player 1 checks Player 2 can check or raise p.
    * If Player 2 checks the betting round ends.
    * If Player 2 raises Player 1 can fold or call.
      · If Player 1 folds Player 2 takes the pot.
      · If Player 1 calls the betting round ends.
  - If Player 1 raises Player 2 can fold or call.
    * If Player 2 folds Player 1 takes the pot.
    * If Player 2 calls the betting round ends.

Showdowns have two possible outcomes: One player has a pair, or both players have the same private card. For the former, the player with the pair wins the pot. For the latter the pot is split.

Kuhn poker has 55 nodes in the game tree and 13 sequences per player. The KJ game tree has 199 nodes, and 57 sequences per player.

To investigate the value that can be derived from exploiting a limited-lookahead opponent, a node evaluation heuristic is needed. In this work we consider heuristics derived from a Nash equilibrium. For a given node, the heuristic value of the node is simply the expected value of the node in (some chosen) equilibrium. This is arguably a conservative class of heuristics, as a limited-lookahead opponent would not be expected to know the value of the nodes in equilibrium. Even with this form of evaluation heuristic it is possible to exploit the limited-lookahead player, as we will show. We will also consider Gaussian noise being added to the node evaluation heuristic, more realistically modeling opponents who have vague ideas of the values of nodes in the game. Formally, let σ be an equilibrium, and i the limited-lookahead player. The heuristic

value $h(s)$ of a node s is:

$$h(s) = \begin{cases} u_i(s) & \text{if } s \in Z \\ \sum_{a \in A_s} \sigma(s,a)h(t_a^s) & \text{otherwise} \end{cases} \quad (16)$$

We consider two different noise models. The first adds Gaussian noise with mean 0 and standard deviation γ independently to each node evaluation, including leaf nodes. Letting μ_s be a noise term drawn i.i.d from $\mathcal{N}(0, \gamma)$: $\hat{h}(s) = h(s) + \mu_s$. The second, more realistic, model adds error cumulatively, with no error on leaf nodes:

$$\bar{h}(s) = \begin{cases} u_i(s) & \text{if } s \in Z \\ \left[\sum_{a \in A_s} \sigma(s,a)\bar{h}(t_a^s)\right] + \mu_s & \text{otherwise} \end{cases} \quad (17)$$

Using the MIP described in the Algorithms section, we computed optimal strategies for the rational player in Kuhn poker and KJ. The MIP models were solved by CPLEX version 12.5. The results are given in Figure 4. The x-axis is the noise parameter γ for the standard deviation in $\hat{h}$ and $\bar{h}$. The y-axis is the corresponding utility for the rational player, averaged over at least 1000 runs for each tuple ⟨game, choice of rational player, lookahead, standard deviation⟩. Each figure contains plots for the limited-lookahead player having lookahead 1 or 2, and a baseline for the value of the game in equilibrium without a limit on lookahead. At each point, the error bars show the standard deviation.

Figures 4a and b show the results for using evaluation function $\hat{h}$ in Kuhn poker, with the rational player in plot a and b being Player 1 and 2, respectively. For rational Player 1, we see that, even with no noise in the heuristic (i.e., the limited-lookahead player knows the value of each node in equilibrium), it is possible to exploit the limited-lookahead player if she has lookahead depth 1. (With lookahead 2 she achieves the value of the game.) For either player and both amounts of lookahead, the exploitation potential steadily increases as noise is added.

Figures 4c and d show the same variants for KJ. Here, lookahead 2 is actually worse for the limited-lookahead player than lookahead 1. To our knowledge, this is the first known incomplete-information *lookahead pathology*. Such pathologies have long been known in perfect-information games [1, 22, 20], and understanding them remains an active area of research [18, 21, 31]. This version of the node heuristic does not have increasing *visibility*: node evaluations do not get more accurate toward the end of the game. Our experiments on KJ with $\bar{h}$ in Figures 4g and h do not have this pathology, and $\bar{h}$ does have increasing visibility.

Perhaps surprisingly, Figure 5 shows a simple subtree (that could be attached to any game tree) where deeper lookahead can make the agent's decision arbitrarily bad, even when the node evaluation function is the exact expected value of a node in equilibrium!

We now go over the example of Figure 5. Assume without loss of generality that all payoffs are positive in some game. We can then insert the subtree in Figure 5 as a subgame at any node belonging to P1, and it will be played with probability 0 in equilibrium, since it has expected value 0. Due to this, all strategies where Player 2 chooses up can be part of an equilibrium. Assuming that P2 is the limited-lookahead player and minimizing, for large enough α, the node labeled P1* will be more desirable than any other node in the game, since it has expected value $-\alpha$ according to the evaluation function. A rational player P1 can use this to get P2 to go down at P2*, and then switch to the action that leads to α. This example is for lookahead 1, but we can generalize the example to work with any finite lookahead depth: the node P1* can be replaced by a subtree where every other leaf has payoff 2α, in which case P2 would be forced to go to the leaf with payoff α once down has been chosen at P2*.

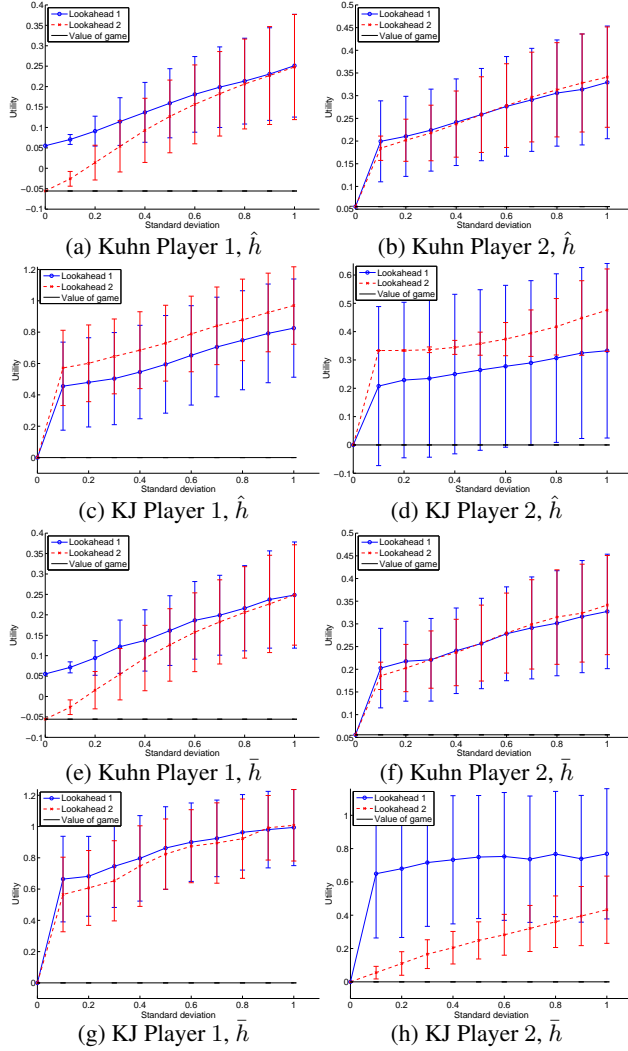Figures 4e and f show the results for Kuhn with $\bar{h}$. These are very

Figure 4: Winnings in Kuhn poker and KJ for the rational player as Player 1 and 2, respectively, for varying evaluation function noise.
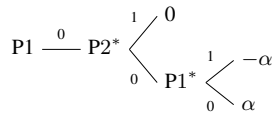

Figure 5: A subtree that exhibits lookahead pathology.

similar to the results for $\hat{h}$, with almost identical expected utility for all scenarios. Figures 4g and h, as previously mentioned, show the results with $\bar{h}$ on KJ. Here we see no abstraction pathologies, and for the setting where Player 2 is the rational player we see the most pronounced difference in exploitability based on lookahead.

7. CONCLUSIONS AND FUTURE WORK

We initiated the study of limited lookahead in incomplete-information games. As a generalization of limited lookahead in complete-information games, we find it interesting in its own right. The game-theoretic reasoning over limited lookahead is another novel aspect. The model also has applications, for example in security games and in steering evolution/adaptation in biomedical games.

We characterized the complexity of finding a Nash equilibrium and optimal strategy to commit to for either player. Figure 6 sum-
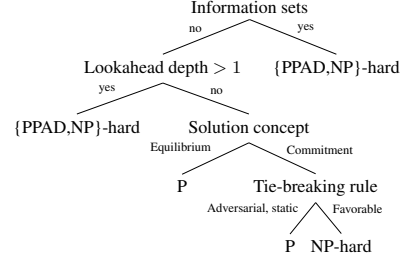
marizes those results.

Figure 6: Our complexity results. {PPAD,NP}-hard indicates that finding a Nash equilibrium is PPAD-hard and finding an optimal strategy to commit to is NP-hard. P indicates polynomial time.

We then designed several MIPs for computing optimal strategies to commit to for the rational player in the general NP-hard cases. First, we showed that the sequence form can be used to design a MIP that has size almost linear in the size of the game tree for many practical games, when ties are broken statically or in favor of the rational player. We then showed that when ties are broken adversarially, the problem reduces to choosing the best among a set of two-player zero-sum games (the tie-breaking being the opponent), and for each of those games the optimal strategy can be computed with an LP. We then introduced a MIP formulation that branches on these games to find the optimal solution.

We experimentally studied the impact of limited lookahead in two poker games. We demonstrated that it is possible to achieve large utility gains by exploiting a limited-lookahead opponent. As one would expect, the limited-lookahead player often obtains the value of the game if her heuristic node evaluation is exact (i.e., it gives the expected values of nodes in the game tree for some equilibrium)—but we provided a counterexample that shows that this is not sufficient in general. Finally, we studied the impact of noise in those estimates, and different lookahead depths. While lookahead 2 usually outperformed lookahead 1, we uncovered an incomplete-information game lookahead pathology: deeper lookahead can hurt the limited-lookahead player. We demonstrated how this can occur with any finite depth of lookahead, even if the limited-lookahead player's node evaluation heuristic returns exact values from an equilibrium.

Our algorithms in the NP-hard adversarial tie-breaking setting scaled to games with hundreds of nodes. For some practical settings, significantly more scalability will be needed. There are at least two exciting future directions toward achieving this. One is to design faster—optimal or good-enough—algorithms. The other is designing abstraction techniques for the limited-lookahead setting. The latter could be used with our current algorithms, or in conjunction with faster future algorithms. In extensive-form game solving with rational players, abstraction plays an important role in large-scale game solving [27]. Theoretical solution quality guarantees have recently been achieved [29, 16, 12, 13]. Limited-lookahead games have much stronger structure, especially locally around an information set, and it may be possible to utilize that to develop abstraction techniques with significantly stronger solution quality bounds. Also, leading practical game abstraction algorithms (e.g., [5]), while theoretically unbounded, could immediately be used to investigate exploitation potential in larger games.

It would also be interesting to explore conditions under which lookahead pathologies occur, and map out similarities and dissimilarities to the pathologies in perfect-information games. Finally, uncertainty over h is an important future research direction. This would lead to more robust solution concepts, thereby alleviating the pitfalls involved with using an imperfect estimate of h.

References

[1] Donald F Beal. "An analysis of minimax". In: *Advances in computer chess* 2 (1980), pp. 103–109.

[2] Bruno Bouzy and Tristan Cazenave. "Computer Go: an AI oriented survey". In: *Artificial Intelligence* 132.1 (2001), pp. 39–103.

[3] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. "Settling the Complexity of Computing Two-Player Nash Equilibria". In: *Journal of the ACM* (2009).

[4] Vincent Conitzer and Tuomas Sandholm. "Computing the Optimal Strategy to Commit to". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. Ann Arbor, MI, 2006.

[5] Sam Ganzfried and Tuomas Sandholm. "Potential-Aware Imperfect-Recall Abstraction with Earth Mover's Distance in Imperfect-Information Games". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2014.

[6] Andrew Gilpin and Tuomas Sandholm. "Lossless Abstraction of Imperfect Information Games". In: *Journal of the ACM* 54.5 (2007).

[7] Samid Hoda et al. "Smoothing Techniques for Computing Nash Equilibria of Sequential Games". In: *Mathematics of Operations Research* 35.2 (2010). Conference version appeared in WINE-07, pp. 494–512.

[8] Michael Johanson et al. "Accelerating Best Response Calculation in Large Extensive Games". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2011.

[9] Daphne Koller and Nimrod Megiddo. "The Complexity of Two-Person Zero-Sum Games in Extensive Form". In: *Games and Economic Behavior* 4.4 (Oct. 1992), pp. 528–552.

[10] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. "Efficient Computation of Equilibria for Extensive Two-Person Games". In: *Games and Economic Behavior* 14.2 (1996), pp. 247–259.

[11] Richard E Korf. "Real-time heuristic search". In: *Artificial intelligence* 42.2 (1990), pp. 189–211.

[12] Christian Kroer and Tuomas Sandholm. "Extensive-Form Game Abstraction With Bounds". In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2014.

[13] Christian Kroer and Tuomas Sandholm. "Extensive-Form Game Imperfect-Recall Abstractions With Bounds". In: *IJCAI workshop on Algorithmic Game Theory*. 2015.

[14] H. W. Kuhn. "A Simplified Two-Person Poker". In: *Contributions to the Theory of Games*. Ed. by H. W. Kuhn and A. W. Tucker. Vol. 1. Annals of Mathematics Studies, 24. Princeton, New Jersey: Princeton University Press, 1950, pp. 97–103.

[15] Marc Lanctot et al. "Monte Carlo Sampling for Regret Minimization in Extensive Games". In: *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2009, pp. 1078–1086.

[16] Marc Lanctot et al. "No-Regret Learning in Extensive-Form Games with Imperfect Recall". In: *International Conference on Machine Learning (ICML)*. 2012.

[17] Joshua Letchford and Vincent Conitzer. "Computing Optimal Strategies to Commit to in Extensive-Form Games". In: *Proceedings of the ACM Conference on Electronic Commerce (EC)*. 2010.

[18] Mitja Luštrek, Matjaž Gams, and Ivan Bratko. "Is real-valued minimax pathological?" In: *Artificial Intelligence* 170.6 (2006), pp. 620–642.

[19] Vahab Mirrokni, Nithum Thain, and Adrian Vetta. "A theoretical examination of practical game playing: lookahead search". In: *Algorithmic Game Theory*. Springer, 2012, pp. 251–262.

[20] Dana S Nau. "Pathology on game trees revisited, and an alternative to minimaxing". In: *Artificial intelligence* 21.1 (1983), pp. 221–244.

[21] Dana S Nau et al. "When is it better not to look ahead?" In: *Artificial Intelligence* 174.16 (2010), pp. 1323–1338.

[22] Judea Pearl. "Heuristic Search Theory: Survey of Recent Results." In: *IJCAI*. Vol. 1. 1981, pp. 554–562.

[23] Judea Pearl. "On the nature of pathology in game searching". In: *Artificial Intelligence* 20.4 (1983), pp. 427–453.

[24] Raghuram Ramanujan, Ashish Sabharwal, and Bart Selman. "On Adversarial Search Spaces and Sampling-Based Planning." In: *ICAPS*. Vol. 10. 2010, pp. 242–245.

[25] Raghuram Ramanujan and Bart Selman. "Trade-Offs in Sampling-Based Adversarial Planning." In: *ICAPS*. 2011, pp. 202–209.

[26] I. Romanovskii. "Reduction of a Game with Complete Memory to a Matrix Game". In: *Soviet Mathematics* 3 (1962), pp. 678–681.

[27] Tuomas Sandholm. "The State of Solving Large Incomplete-Information Games, and Application to Poker". In: *AI Magazine* (2010). Special issue on Algorithmic Game Theory, pp. 13–32.

[28] Tuomas Sandholm. "Steering Evolution Strategically: Computational Game Theory and Opponent Exploitation for Treatment Planning, Drug Design, and Synthetic Biology". In: *AAAI Conference on Artificial Intelligence (AAAI)*. Senior Member Track. 2015.

[29] Tuomas Sandholm and Satinder Singh. "Lossy stochastic game abstraction with bounds". In: *Proceedings of the ACM Conference on Electronic Commerce (EC)*. 2012.

[30] Bernhard von Stengel. "Efficient Computation of Behavior Strategies". In: *Games and Economic Behavior* 14.2 (1996), pp. 220–246.

[31] Brandon Wilson et al. "Improving Local Decisions in Adversarial Search." In: *ECAI*. 2012, pp. 840–845.

[32] Z Yin et al. "TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems using Game Theory". In: *AI Magazine* (2012).

[33] Martin Zinkevich et al. "Regret Minimization in Games with Incomplete Information". In: *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2007.