Discretization of Continuous Action Spaces in Extensive-Form 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

Extensive-form games are a powerful tool for modeling a large range of multiagent scenarios. However, most solution algorithms require discrete, finite games. In contrast, many real-world domains require modeling with continuous action spaces. This is usually handled by heuristically discretizing the continuous action space without solution quality bounds. In this paper we address this issue. Leveraging recent results on abstraction solution quality, we develop the first framework for providing bounds on solution quality for discretization of continuous action spaces in extensive-form games. For games where the error is Lipschitz-continuous in the distance of a continuous point to its nearest discrete point, we show that a uniform discretization of the space is optimal. When the error is monotonically increasing in distance to nearest discrete point, we develop an integer program for finding the optimal discretization when the error is described by piecewise linear functions. This result can further be used to approximate optimal solutions to general monotonic error functions. Finally we discuss how our theory applies to several practical problems for which no solution quality bounds could be derived before.

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

Extensive-form games; Nash equilibrium; equilibrium finding; discretization; continuous action spaces

1. INTRODUCTION

Game-theoretic equilibrium concepts play an increasingly important role in prescribing how agents should act in multiagent settings. The problem of computing Nash equilibria and related concepts has received significant attention in the literature (e.g., [28, 27, 12, 18]). Extensive-form games are an especially powerful class of games, that model many sequential real-world domains, such as sequential auctions, negotiation, security domains, cybersecurity, medical applications of game theory, and recreational games such as poker and billiards [1, 34, 29, 8, 9, 35].

The computational literature has largely focused on games where the action spaces are discrete at every node of the game. Indeed, most algorithms for solving extensive-form games require this (e.g. [39, 42, 25, 17, 32]). (One notable exception to this was introduced by Johanson et al. [20], where a technique was demonstrated for handling continuous action spaces for nature in a fairly restricted sense.) However, not all games encountered in practice are discrete. Sources of continuity include noise (e.g. Gaussian) being modeled by a nature node, bid sizes in auctions, betting sizes in poker games, type profiles in mechanism design, and power levels in jamming games.

In the past, such continuity has largely been handled by heuristic discretization, with no theoretical guarantees on solution quality. In contrast, we develop the first general bounds on solution quality for discretizations of continuous action spaces in a very broad class of games. Building on recent results on abstraction in extensive-form games [23], we show how to discretize continuous action spaces in a way that gives theoretical bounds on solution quality when using any Nash equilibrium computed in the discretized game to form a strategy in the full (continuous) game. This is highly nontrivial because abstraction in games is *pathological* unlike in single-agent settings such as MDPs: there are cases where the (best) equilibrium strategy computed from a fine-grained abstraction is worse (more exploitable) than the (worst) equilibrium computed from a coarser abstraction, when measured on the original unabstracted game [40]!

We then proceed to investigate the computation of discretizations that minimize an error bound. We first formulate the general problem, making only the assumption that the error bound function is *monotonic*, which intuitively requires that the error gets worse the farther a point gets from the discrete point to which it maps. Since our problem formulation consists of linear constraints, this immediately leads to the conclusion that convex error functions can typically be minimized in polynomial time. We then go on to consider error functions of the form derived in our theoretical solution-quality bounds. We show that when individual leaf and nature node error functions are linear, and nature always picks uniformly over continuous action intervals, the

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

bound-minimizing solution is to uniformly discretize each continuous interval. We further show how to decompose the problem and we develop a general framework for optimizing the decomposed formulation for convex functions. We then develop a mixed-integer program (MIP) for computing optimal discretizations when only player action spaces are being discretized and the error functions are piecewise linear.

Our paper is similar in spirit to two recent strands of research in game solving. First is the bounded abstraction literature. Our bounds leverage recent results in game abstraction [23]. In addition to that work, solution-quality bounds have also been obtained for imperfect-recall games. Lanctot et al. [26] give solution-quality bounds for running the counterfactual regret minimization algorithm on an imperfect-recall abstraction of a perfect-recall game. Kroer and Sandholm [24] give solution-quality bounds on implementing Nash equilibria computed from imperfect-recall abstractions in the original perfect-recall game. Both sets of results allow only information abstraction, not action removal, and thus cannot discretize continuous actions. Gilpin and Sandholm [12] give lossless abstraction results and algorithms. These are for a specialized class of games called *games of ordered signals*, and apply only to finite-sized games. Similarly, Basilico and Gatti [2] give bounds for another finite specialized class of games called *Patrolling Security Games*. There have been several recent papers on evaluating the quality of an abstraction after the fact—typically for games with discrete action spaces [13, 19, 20, 21]. This is not feasible in most continuous games, and also our goal is to devise good abstractions up front rather than evaluating strategies computed using abstraction after the fact. Sandholm and Singh [36] give solution quality bounds for stochastic games. For the special case of our results where the game is a stochastic game, their results could also be applied, similarly to how we apply the results of Kroer and Sandholm [23]. However, the latter lead to stronger bounds.

Second is the literature on practical game abstraction, which has focused on poker. This literature is less related to our work, as our goal is to derive explicit general bounds, rather than experimentally constructing abstractions that work well for a particular game. The experimental game-specific approach could, of course, potentially yield discretizations with better bounds because specific games can have additional structure. Initially, game abstractions were created by hand, using domain dependent knowledge [38, 4]. More recently, automated abstraction has taken over [11, 12, 42]. This has typically been used for information abstraction, whereas action abstraction is still largely done by hand [14, 37]. Recently, automated action abstraction approaches have also started to emerge [15, 16, 6]. In addition to focusing on theoretical solution quality bounds, our work also does not distinguish between information and action abstraction, as we consider both discretization of nature (information) and player nodes.

2. EXTENSIVE-FORM GAMES

Extensive-form games provide a general framework for modeling scenarios where agents make decisions sequentially (or simultaneously), potentially with imperfect information about the state of the game. Here we give a short formal description. For a complete description see Osborne and Rubinstein [31]. An extensive-form game is a tuple $\Gamma = \langle N, A, S, Z, \mathcal{H}, \sigma_0, u, \mathcal{I} \rangle$. 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 form a tree with root node $r \in S$. At each node s, it is the turn of some Player $P(s)$ who gets to choose an action from action set A_s. Each branch at s denotes a different choice in A_s. The set of all nodes where Player i is active is denoted by $S_i = \{s \in S : P(s) = i\}$. $Z \subset S$ 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 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.

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

Perfect recall means that no player forgets anything that that player observed. Formally, for every Player $i \in N$, information set $I \in \mathcal{I}_i$, and nodes $s_1, s_2 \in I$, the sequence of actions performed by Player i is the same for s_1 and s_2.

We denote by σ_i a *behavioral strategy* for Player i. For each information set I where it is the player's turn to move, it assigns a probability distribution over the actions A_I 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 uniquely 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. We let $\sigma_{I \to a}$ denote the strategy profile obtained from σ by having Player i deviate to taking action a at $I \in \mathcal{I}_i$. The probability of reaching node s is $\pi^\sigma(s) = \Pi_{\{\bar{s}, \bar{a}\} \in X_s} \sigma(\bar{s}, \bar{a})$ where X_s is the set of pairs of nodes and actions on the path from the root to s. $\pi^\sigma(I) = \sum_{s \in I} \pi^\sigma(s)$ is the probability of reaching any node in I. 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, but without Player i. Given a strategy profile σ, we let $u_i(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u_i(z)$ denote the expected utility of player i under σ. We let $u_i(\sigma_i, \sigma_{-i})$ denote the same quantity, allowing us to swap the strategy for player i.

We denote by t_a^s be the node transitioned to by performing action $a \in A_s$ at any node s.

2.1 Continuous action spaces

We will assume that we are dealing with a game Γ, where one or more nodes $s \in S$ have one or more continuous action intervals $A_{s,c} = [\alpha_{s,c}, \beta_{s,c}] \subseteq A_s$ for $c \in C_s$, where C_s is an index set of the continuous action spaces at s.

Let S_a be the set of nodes in the subtree reached by taking action $a \in A_{s,c}$ at node s. We assume there is a one-to-one mapping $\phi_{a,\hat{a}} = \phi_{\hat{a},a}$ between S_a and $S_{\hat{a}}$ for any two actions $a, \hat{a} \in A_{s,c}$, where nodes at a given height map onto nodes at the same height, and the condition of Definition 1 is satisfied. Intuitively, the condition requires that nodes that are mapped to each other are either (1) in the same information set, or (2) their information sets contain no nodes from outside the (infinitely large) set of subtrees reachable by taking

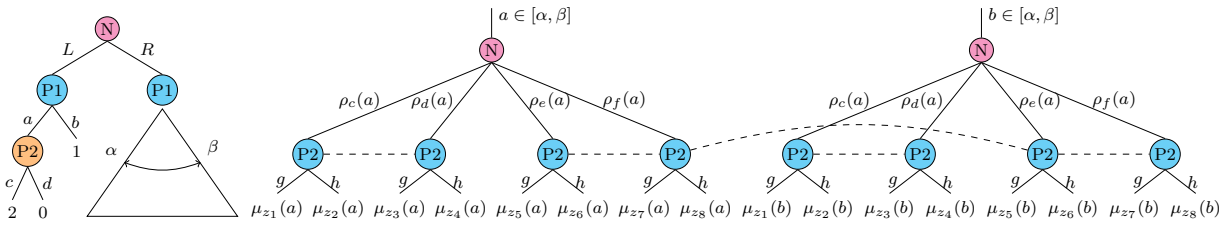

Figure 1: An example of a game tree with the triangle representing a subtree with a continuous action space at the node.

actions from $A_{s,c}$ at s, and for any other node in the same information set and same subtree, the nodes mapped to must similarly be in the same information set.

Definition 1. *For any node $\hat{s} \in S_a$ in the subtree at $a \in A_{s,c}$, we require one of the two following conditions to hold for all a':*

1. $\phi_{a,a'}(\hat{s}) \in I_{\hat{s}}$
2. $I_{\hat{s}} \subset \bigcup_{\bar{a} \in [\alpha_{s,c}, \beta_{s,c}]} S_{\bar{a}}$, and for any other node $\bar{s} \in I_{\hat{s}}, \bar{s} \in S_a$: $\phi_{a,a'}(\bar{s}) \in I_{\phi_{a,a'}(\hat{s})}$

For each leaf node $z \in Z_{t_a^s}$ for some $a \in A_{s,c}$, we assume that the payoff to Player i can be described as $u_i(z) = \mu_z^i(a)$, where $\mu_z^i = \mu^i_{\phi_{a,\hat{a}}(z)}$ for all $\hat{a} \in A_{s,c}$. Intuitively, all leaf nodes that map to each other have their payoffs described by the same function, with the actual value depending on the choice of a. Similarly, the probability of each outcome at each nature node s' in the subtree rooted at t_a^s is described by a function $\rho_{s'}(a)$ where $\rho_{s'} = \rho_{\phi_{a,\hat{a}}(s')}$ for all $\hat{a} \in A_{s,c}$.

Since we require a bijection mapping $\phi_{a,\hat{a}}$ between the subtrees for any two actions $a, \hat{a} \in [\alpha_{s,c}, \beta_{s,c}]$ for each node s with continuous action interval c, the infinitely many subtrees can equivalently be thought of as infinitely many instantiations of a single game-tree prototype, where payoffs and nature outcome probabilities are parameterized by the choice of $a \in [\alpha_{s,c}, \beta_{s,c}]$, and the information set topology of the instantiations satisfy Definition 1.

An example game is shown in Figure 1. On the left is a game with three players: nature(N), Player 1 ($P1$), and Player 2 ($P2$). Nature first chooses between L and R (with some unspecified fixed probability distribution). If L is chosen, a discrete subgame is reached. If R is chosen, $P1$ has a continuous action space $[\alpha, \beta]$. In the middle and right are shown two specific subtrees under the continuous action space, for $P1$ choosing actions $a, b \in [\alpha, \beta]$, respectively. The information set that spans across the trees is an example of one that satisfies Condition 1 of Definition 1, while the one that does not span across the trees satisfies Condition 2.

So far we have described our notation in terms of discretizing a single continuous action space at some node. When there is more than one node with a continuous action space that is being discretized, we have to consider two groups of nodes with continuous action spaces.

The first group is the set of all nodes s that have one or more continuous action intervals, and s is the first node on the path from the root to s with a continuous action space. Let the set of all such nodes be $\mathcal{S}^1 \subseteq S$. These nodes are handled exactly as described above. If there are additional continuous action spaces in the subtree after taking some action $a \in A_{s,c}$ for some s, c, then the bijections between subtrees simply map uncountably many nodes.

The second group is the set of all nodes s' such that there is at least one node-action pair s, a on the path from the root to s', where s has a continuous action space $A_{s,c}$ such that $a \in A_{s,c}$. Let this set of nodes be called $\mathcal{S}^2 \subset S$. Let $\vec{a} \in \mathbb{R}^n$, where n is the number of continuous action spaces leading to s' and including the one at s', such that $a_i \in [\alpha_i, \beta_i]$ where $[\alpha_i, \beta_i]$ is the continuous action space for the i'th node with a continuous action space on the path to s'. We then require that the payoff and nature functions are functions of $\vec{a}$ rather than a single action choice. For fixed choices in the past, the functions then behave exactly as the functions for nodes in $\mathcal{S}^1$.

We fix some ordering of all continuous intervals, and define $\mathcal{A} = \times_{s \in \mathcal{S}^1 \cup \mathcal{S}^2, c \in C_s} A_{s,c}$ to be the Cartesian product over all continuous intervals. We will use j to denote indices into this ordering $\mathbb{I}$. We let $[\alpha_j, \beta_j]$ denote the endpoints of interval $j \in \mathbb{I}$. From now on we will use $\vec{a} \in \mathcal{A}$ to denote elements of this set. A *discretization* is a finite set of points $\mathcal{A}' \subset \mathcal{A}$. The size $|\mathcal{A}'| = m$ is the number of discrete points chosen for each interval. If fewer than m points are desired for some interval $A_{s,c}$ with index $j \in \mathbb{I}$, we can simply let $\vec{a}_j = \vec{a}'_j$ for distinct points $\vec{a}, \vec{a}' \in \mathcal{A}'$. For a node s, we will use $\vec{a}^s$ to refer to the subset of actions taken on the path to s such that the action is part of a continuous action interval. We will overload the notation of the payoff and nature error functions so that for a given f or h for a node s, they take elements $\vec{a} \in \mathcal{A}$ as input, with the implicit understanding that the value of the function depends only on $\vec{a}^s$. We will let $\pi_0(\vec{a})$ denote the product of probabilities over each index j into $\vec{a}$ such that nature acts at $\vec{a}_j$.

The game $\Gamma' = \langle N, A', S', Z', \mathcal{H}, \sigma_0, u, \mathcal{I}' \rangle$ is the discrete extensive-form game obtained by restricting players to selecting actions that are part of the discretization $\mathcal{A}'$.

2.2 Equilibrium notions

In this work, the kind of strategy profiles that we will focus on are ones that constitute an (approximate) *Nash equilibrium*. A Nash equilibrium is a strategy profile σ such that for each agent i and alternative strategy σ_i': $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i})$. An *$\epsilon$-Nash equilibrium* is a strategy profile where each agent can gain at most ϵ by deviating, as long as the strategies of the other agents are kept fixed. Formally, for each agent i and alternative strategy σ_i': $u_i(\sigma_i, \sigma_{-i}) + \epsilon \geq u_i(\sigma_i', \sigma_{-i})$. In a Nash equilibrium, $\epsilon = 0$.

2.3 Discretization mapping and error terms

We will need to reason about the similarity of the full game Γ and the induced game Γ' for a given discretization $\mathcal{A}'$. To do this we will require a mapping of the continuous

action space onto the discretization:

DEFINITION 2. *A discretization mapping is a surjective function $g : \mathcal{A} \to \mathcal{A}'$ that maps the continuous action space $\mathcal{A}$ onto the discretization $\mathcal{A}'$. We require that g is decomposable, so for all $\vec{a} \in \mathcal{A}$, $g(\vec{a})_j$ depends only on $\vec{a}_j$. $\mathcal{G}_{\mathcal{A}'}$ denotes the set of legal discretization maps for a given discretization $\mathcal{A}'$.*

The discretization mapping along with the bijections $\phi_{a,a'}$ for all $s \in S, c \in C_s, a, a' \in A_{s,c}$ immediately defines a mapping of the nodes S onto the nodes S'. Denote this node mapping function by $h : S \to S'$. For any node $s \in S \cap S'$, $h(s) = s$. For $s \in S, s \notin S'$, $h(s)$ is the node in S' reached by inductively applying the maps g and $\phi_{\vec{a}^s_j, g(\vec{a}^s)_j}$ at each continuous action space on the path to s.

Due to the constraints in Definition 1, g also leads to an information set mapping, as any two nodes $s_1, s_2 \in I$ for some I must map to the same information set: $h(s_1), h(s_2) \in I'$ for some I'. We let $f : \mathcal{I} \to \mathcal{I}'$ be this information set mapping.

For all three functions g, h, f we also define their inverses g^{-1}, h^{-1}, f^{-1}, that return all intervals $\bar{A} \subseteq \mathcal{A}$, nodes $\bar{S} \subset S$, and information sets $\bar{\mathcal{I}} \subseteq \mathcal{I}$, respectively, that map onto given $\vec{a} \in \mathcal{A}', s' \in S'$, and $I' \in \mathcal{I}'$, respectively. We denote by $h_I^{-1}(s')$ the intersection $h^{-1}(s') \cap I$.

Given a discretization mapping g, it will be convenient to define aggregate utility error terms for nodes of the real game:

$$\epsilon^R_{s,i} = \begin{cases} \max_{a \in A_s} \epsilon^R_{t^s_a,i} & \text{if } s \text{ is a player node} \\ \int_{a \in A_s} \sigma_0(s,a) \epsilon^R_{t^s_a,i} & \text{if } s \text{ is a nature node} \\ \left|\mu^i_s(\vec{a}) - \mu^i_s(g(\vec{a}))\right| & \text{if } s \text{ is a leaf node} \end{cases}$$

Similarly, we define aggregate error terms for nature error. We define the nature distribution error of an information set I and node $s' \in f(I)$ to be

$$\epsilon^0_{I,s'} = \left| \frac{\int_{s \in h_I^{-1}(s')} \sigma_0(s)}{\sigma_0(I)} - \frac{\sigma'_0(s')}{\sigma'_0(f(I))} \right|$$

This is the difference between nature's probability of reaching s' and its probability of reaching any node in $h_I^{-1}(s')$, normalized by the probability of reaching the given information sets. The nature error for information set I is

$$\epsilon^0_I = \sum_{s' \in f(I)} \epsilon^0_{I,s'}$$

For a nature node s at height $k \in \mathcal{H}_0$ and $s' = h(s)$, we define the nature action $a' \in A_{s'}$ error and node error to respectively be

$$\epsilon^0_{s,s',a'} = \left| \sigma'_0(s', a') - \int_{a \in g^{-1}(a') \cap A_s} \sigma_0(s,a) \right|$$
$$\epsilon^0_s = \sum_{a' \in A_{s'}} \epsilon^0_{s,s',a'}$$

The nature error at height k is

$$\epsilon^0_k = \begin{cases} \int_{I \in \mathcal{I}_k} \pi(I) \epsilon^0_I & \text{if } k \notin \mathcal{H}_0 \\ \int_{s \in S_k} \pi(s) \epsilon^0_s & \text{if } k \in \mathcal{H}_0 \end{cases}$$

Finally, we let $\bar{W} = \max_{i \in N, z \in Z} u_i(z)$ be the maximum payoff at any leaf node.

2.4 Strategy mapping

Once a Nash equilibrium has been computed in the discretized game, we need to define a way of converting that strategy profile to a strategy profile for the full game. We perform this mapping in the following simple way. Since all subtrees under discretized intervals have the same shape (based on how we defined the discretization problem in Section 2.1) and thus same number of actions, we can simply implement the same strategy that we computed for a given discretized subtree at all subtrees that map to it. Specifically, let σ' be the strategy profile computed for the discretized game. Then for each real node $s \in S$, we set $\sigma(s,a) = \sigma'(h(s), a')$, where a' is the action at s' that leads to $h(t^s_a)$. For continuous intervals, we simply pick each discrete point a with the probability that it was chosen in the discrete game, and every other action with probability zero. This mapping yields a strategy profile that satisfies the following property, which we will use to derive bounds later:

PROPOSITION 1. *For a strategy profile σ' computed in a discretized game Γ', our method of strategy conversion leads to a strategy profile σ for the full game Γ so that for any information set pair I, I' such that I maps onto I', $\sigma_{-0}(I) > 0$, and $\sigma_i(I) > 0$,*

$$\left| \frac{\sigma(s')}{\sigma(I')} - \sum_{s \in g_I^{-1}(s')} \frac{\sigma(s)}{\sigma(I)} \right| \leq \epsilon^0_{I,s'}$$

PROOF. This follows immediately from how we defined the mapping. □

3. OVERVIEW OF OUR APPROACH

Given some game with continuous action spaces, the goal in this paper is to pick a finite set of points for each continuous action interval. This will induce a finite extensive-form game. A (potentially approximate) Nash equilibrium is then computed in the discrete game. The computed Nash equilibrium is then mapped to a strategy profile in the full (continuous) game. Figure 2 illustrates the approach.

Under reasonable assumptions, we will derive solution quality bounds for any Nash equilibrium computed in the abstraction when implemented in the full game. More specifically, we will show that such strategy profiles constitute ϵ-Nash equilibria in the full game, where the ϵ depends on the error terms we defined in the previous section. These results are anlogous to the solution-quality results for discrete games [23, 24].

4. DISCRETIZATION QUALITY BOUNDS

We start by showing an error bound on solution quality for any given discretization and discretization mapping, leveraging a recent result of Kroer and Sandholm [23].

THEOREM 2. *For any game Γ with continuous action spaces $\mathcal{A}$ that satisfy the constraint given in Definition 1, discretization $\mathcal{A}' \subset \mathcal{A}$, and discretization mapping $g : \mathcal{A} \to \mathcal{A}'$, any Nash equilibrium σ computed in Γ' constitutes an ϵ-Nash equilibrium when implemented in Γ, where*

$$\epsilon = \max_i \left\{ 2\epsilon^R_{s,i} + \sum_{k \in \mathcal{H}_i} \epsilon^0_k \overline{W} \right\} + 2 \sum_{k \in \mathcal{H}_0} \epsilon^0_k \overline{W}$$

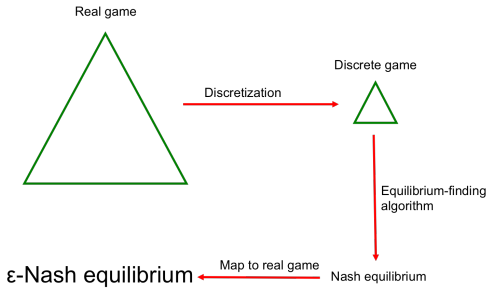

Figure 2: An overview of our discretization approach.

PROOF. We can view the game Γ' obtained by this discretization as an abstraction, where g defines a mapping of each real action a to some abstract action a'. Coupled with the bijection $\phi_{a,a'}$, this induces a node mapping h and information set mapping f, as argued in Section 2.3.

Theorem 4.2 of Kroer and Sandholm [23] gives error bound results for abstractions like this. To see that f indeed forms a surjective function that respects h, consider Conditions 1 and 2 of Definition 1, which ensure that any discretization induces an information set mapping. First consider when Condition 1 is satisfied. In this case the information set mapping is obviously respected, as the nodes that map to each other were already in the same information set.

Now consider any information set I such that a node $s \in I$ is mapped onto another node $\hat{s} \in \hat{I}, \hat{I} \neq I$. Condition 2 ensures that this only happens for information sets completely contained in the subtrees rooted at the continuous interval in question. We have to show that I is surjectively mapped onto $\hat{I}$. Condition 2 states that for any other node $\bar{s} \in I$, $\bar{s}$ must map to a node in the same information set as $\hat{s}$. Thus we just have to verify that every node $s^* \in \hat{I}$ has a node from I mapped onto it. This is immediately seen by applying Definition 1 to the bijection from the perspective if $\hat{I}$, as the bijection has to be the same when applying the definition in both directions.

Theorem 4.2 of Kroer and Sandholm [23] also requires that the strategy implemented in the full game is an *undivided lifted strategy*. We did not quite guarantee this property with our strategy mapping described in Section 2.4. However, this property is only used in the very last step of the proof of Theorem 4.2 in their paper, where they use it to apply Proposition 3.1 of their paper. Instead, we can apply our Proposition 1, which achieves the same effect.

The result as presented by Kroer and Sandholm [23] takes the maximum nature outcome error for each height. Their proof is easily modified to take the expected value, ϵ_k^0, instead, as we do in the theorem. (This approach was also taken by Kroer and Sandholm [24].) □

The bounds as given here are in their most general form. In particular, the two nature error terms $\sum_{k \in \mathcal{H}_i} \epsilon_k^0 \overline{W}$ and $2\sum_{k \in \mathcal{H}_0} \epsilon_k^0 \overline{W}$ are not given in terms of the functions ρ_s that describe the change in nature outcome probabilities. ϵ_k^0 can easily be bounded for all k in $\mathcal{H}_i$ and $\mathcal{H}_0$ respectively:

$$\epsilon_k^0 \leq \int_{I \in \mathcal{I}_k} \pi(I) \sum_{s' \in f(I)} \left| \int_{s \in h_I^{-1}(s')} \pi_0(\vec{a}^s) - \pi_0(\vec{a}^{s'}) \right| = \xi_k^0$$

$$\epsilon_k^0 \leq \int_{s \in S_k} \pi(s) \sum_{a' \in A_{s'}} \left| \int_{a \in g_I^{-1}(a')} \sigma_0(s,a) - \sigma_0(s',a') \right| = \xi_k^0$$

This gives the following corollary:

COROLLARY 3. *For any game Γ with continuous action spaces $\mathcal{A}$ that satisfy the constraint given in Definition 1, discretization $\mathcal{A}' \subset \mathcal{A}$, and discretization mapping $g : \mathcal{A} \to \mathcal{A}'$, any Nash equilibrium σ computed in Γ' constitutes an ϵ-Nash equilibrium when implemented in Γ, where*

$$\epsilon = \max_i \left\{ 2\epsilon_{s,i}^R + \sum_{k \in \mathcal{H}_i} \xi_k^0 \overline{W} \right\} + 2 \sum_{k \in \mathcal{H}_0} \xi_k^0 \overline{W}$$

The ξ_k^0 terms do not diverge. While an infinite sum is taken in both cases, the terms are probability weighted, and $\int_{I \in \mathcal{I}_k} \pi(I) = 1, \int_{s \in S_k} \pi(s) = 1$. Since we are dealing with probability distributions, we can take the maximum over $\mathcal{I}_k$ or S_k. In practical settings, it may be desireable to take several maxima. First, in the current form of both Theorem 2 and Corollary 3, the bound depends on the strategy profile of the players, not just nature. To make the bound independent of player actions, one can take the maximum over player actions. Second, instead of computing the infinite sum of errors over $\mathcal{I}_k$ or S_k, it may be useful to take the probability-weighted sum of errors over discrete points, and then compute the maximum error over each discrete point.

5. DISCRETIZATION ALGORITHMS

In this section we consider general bounded discretization problems that are largely independent of the specific error bound to be minimized. This means that our algorithms will apply to the results from Section 4, and also to any (potentially stronger or more general) bounds obtained in the future, as long as they fall under the setting described in Section 5.1.

5.1 The optimization problem

We consider a more general class of problems than those described in Section 2.1. We consider a game Γ where one or more $s \in S$ has a continuous action space. We again let $\mathcal{A}$ be the set of all intervals to be discretized with index set $\mathbb{I}$ and let $\vec{a}_j^l$ refer to the k_j discrete points chosen for interval j. We assume the points in the discretization are ordered, so $\vec{a}_j^l < \vec{a}_j^{l+1}$ for all $j \in \mathbb{I}, l \in [k_j]$.

We start by formulating the optimization problem very generally. We assume that we have an error bounding function $\Psi : \mathcal{A}^k \times \mathcal{G}_\mathcal{A} \to \mathbb{R}$ that takes as input a discretization $\mathcal{A}'$ and discretization map g and returns a real-valued error bound $\Psi(\mathcal{A}', g)$. We make two innocuous assumptions about Ψ to represent the natural condition that the error increases the further an actual point is from the discrete point to which it maps.

First, each point maps to its nearest lower or upper discrete point. Formally, for all $j \in \mathbb{I}$, and any point a_l, $\vec{a}_j^l < a_l < \vec{a}_j^{l+1}$ not in the discretization, Ψ is minimized at $g(a_l) = \vec{a}_j^l$ or $g(a_l) = \vec{a}_j^{l+1}$.

Second, for all $j \in \mathbb{I}$, and any two points $a_l, \hat{a}_l$, $\vec{a}_j^l < a_l < \hat{a}_l < \vec{a}_j^{l+1}$ not in the discretization, if Ψ is minimized when $g(\hat{a}_l) = \vec{a}_j^l$ then Ψ is minimized when $g(a_l) = \vec{a}_j^l$, and if Ψ is minimized when $g(a_l) = \vec{a}_j^{l+1}$ then Ψ is minimized when $g(\hat{a}_l) = \vec{a}_j^{l+1}$.

We will say that an error function Ψ that satisfies our two assumptions is *monotonic*. Given a monotonic Ψ and a discretization $\mathcal{A}'$, the optimal mapping for each interval $j \in \mathbb{I}$ can be determined by finding the splitting point between each interval $\left[\vec{a}_j^l, \vec{a}_j^{l+1}\right]$ such that the left side of the splitting point is mapped onto $\vec{a}_j^l$ and right side is mapped onto $\vec{a}_j^{l+1}$.

For each interval $j \in \mathbb{I}$, we introduce real-valued variables $\vec{a}_j^l \in \mathcal{A}_j, l \in [k_j]$ and $g_{s,c}^\tau, \tau \in [k_j - 1]$, where k_j is the desired number of discrete points for interval $\mathcal{A}_j$. The variables $\vec{a}_j^l$ represent the discrete points, while g_j^τ represents the point between $\vec{a}_j^\tau$ and $\vec{a}_j^{\tau+1}$ that separates the interval, such that the points in the interval $\left[\vec{a}_j^\tau, g_j^\tau\right]$ map onto $\vec{a}_j^\tau$ and the points in the interval $\left[g_j^\tau, \vec{a}_j^{\tau+1}\right]$ map onto $\vec{a}_j^{\tau+1}$. Since any set of values for $\vec{a}_j^l, g_j^\tau$ over all $j \in \mathbb{I}, l \in [k_j], \tau \in [k_j - 1]$ completely specifies a discretization and discretization mapping, we let $\mathcal{A}'_v, g_v$ denote the discretization and mapping obtained by a solution. The feasible set of this problem is

$$\mathcal{F} = \left\{ \mathcal{A}'_v, g_v : \begin{array}{ll} \vec{a}_j^l \le \vec{a}_j^{l+1} & \forall j \in \mathbb{I}, l \in [k_j - 1] \\ \vec{a}_j^\tau \le g_j^\tau \le \vec{a}_j^{\tau+1} & \forall j \in \mathbb{I}, \tau \in [k_j - 1] \\ \vec{a}_j^l \in \mathcal{A}_j & \forall j \in \mathbb{I}, l \in [k_j] \\ g_j^\tau \in \mathcal{A}_j & \forall j \in \mathbb{I}, \tau \in [k_j - 1] \end{array} \right\} \quad (1)$$

With this notation, we arrive at the most generic form of the optimal discretization problem for monotonic error functions:

$$\min \left\{ \Psi(\mathcal{A}'_v, g_v) : \left(\mathcal{A}'_v, g_v\right) \in \mathcal{F} \right\} \quad (2)$$

Setting $k = 1$, one can see that this is equivalent to minimizing *any* function, so this general form will not get us far. Fortunately, there is often further structure in practical games. In the rest of the algorithms section, we will consider various forms of structure that enable us to design efficient algorithms for finding good discretizations.

5.1.1 Convex error function

The constraints specified in (1) are all linear. Thus, if Ψ is convex, solving (2) becomes a convex minimization problem over linear constraints. These are solvable in polynomial time under mild assumptions [3][1]. Perhaps more importantly, large subsets of this class of error functions have practically efficient solution methods—e.g., an error function that is conic-quadratic or semi-definite representable (Ben-Tal and Nemirovski [3] give a thorough discussion of such representability [1]), smooth, or non-smooth with certain structure [30].

5.2 Decomposable error function

In Section 4, we considered error functions that are a mixture of maximums (player nodes) and probability-weighted sums (nature nodes) of the error functions at individual nodes. We now consider how to (recursively) represent such error functions using linear inequalities, for the purpose of computation.

Let $\xi_j : \mathcal{A}_j \times g_j \to \mathbb{R}$ be an error function that gives the error incurred at interval $j \in \mathbb{I}$ when choosing discretization $\mathcal{A}'_j$ and mapping g_j at $\mathcal{A}_j$. Recursively taking the maximum or weighted sum can be implemented by recursively applying the following linear inequalities, where variable e_δ represents the error at a given node or information set δ:

$$\mathcal{E} = \left\{ \begin{array}{l} e_s \ge \sum_{a \in A_s} \sigma_0(s,a) e_{t_a^s} \\ e_s \ge \max_{a \in A_s} e_{t_a^s} \\ e_s \ge \xi_j \end{array} \right\} \quad (3)$$

The same linear formulation can also be used to formulate the error bound in the case where $\xi_s : \mathcal{A} \times g$ are functions that give the error for each leaf and nature node. In a slight abuse of notation, we denote the set of error function solutions described in (3) by $\mathcal{E}$.

If each error function ξ_j depends only on the subset of $\mathcal{A}$ that consists of interval j and any descendant intervals, the discretization problem can be decomposed: find each interval $j \in \mathbb{I}$ that is the first continuous interval from the root to the interval. Each such interval, along with its subtrees and any intervals therein, can be minimized separately.

5.3 Minimizing our bound

We will now study the game class that satisfies Definition 1 and minimization of the bound given in Corollary 3. We will consider various types of error functions.

5.3.1 Linear error functions

In this section we consider games Γ where the utility and nature outcome distribution functions μ_z, ρ_s, at all $z \in Z, s \in S$ that are descendants of a continuous interval, are Lipschitz continuous with Lipschitz constant $L_{z/s}$. This encompasses two important practical classes of game: (1) all the functions μ and ρ are linear, and (2) the functions are non-linear but the bound being derived is based on knowing (only) that the functions are Lipschitz continuous.

If all continuous intervals at nature nodes have a uniform distribution, we get the following simple results: all intervals should be discretized into uniform interval sizes.

THEOREM 4. *For a game Γ with continuous action spaces $\mathcal{A}$, where each utility and nature outcome distribution function is Lipschitz continuous with constant $L_{z/s}$ for each leaf z or nature node s, and all continuous nature intervals are uniformly distributed, the bound-minimizing k_j-point discretization at each interval $\mathcal{A}'_j$ is:*

$$\vec{a}_j^l = \alpha_j + \left(l - \frac{1}{2}\right) \cdot \left(\frac{\beta_j - \alpha_j}{k_j}\right) \qquad \forall l \in [k_j]$$

$$g_j^\tau = \alpha_j + \tau \cdot \left(\frac{\beta_j - \alpha_j}{k_j}\right) \qquad \forall \tau \in [k_j - 1]$$

We will call this a uniform discretization.

PROOF. Consider such a discretization and mapping. We will show that any other discretization or mapping can not be better. For an interval $j \in \mathbb{I}$ at a player node, this is easily seen. The error of the interval is the maximum error over the interval. Since all functions are linear, this is simply the point $a \in \mathcal{A}_j$ with the largest distance to its discrete point. For a uniform discretization, this is either endpoint α_j, β_j or some inner point g_j^τ, which all have distance $\frac{\beta_j - \alpha_j}{2k_j}$. For any other discretization, some point must have distance strictly greater than this to its discrete point, thus worsening the bound.

[1] A heavily and continuously updated version of this book is at `http://www2.isye.gatech.edu/~nemirovs/`.

For an interval $j \in \mathbb{I}$ at some nature node s, we first observe that the error function is convex. The error at each leaf or nature node in the subtrees is linear (and thus convex). The error at each other node is the finite sum or maximum over descendant errors. The error over the interval is the integral over error at each subtree. Since taking finite sums, maxima, and integrals all preserve convexity (see Boyd and Vandenberghe [5] for a calculus of convex functions), we get that the error over the interval is convex. For convex functions, local minimizers are global minimizers. Thus, it is sufficient to show that the derivative is zero at the uniform discretization. Since the subtrees have the same structure, by the conditions given in Definition 1, and the fact that the error functions depends only on the distance, we get that the error between any two points $a_1, a_2 \in \mathcal{A}_j$ can be represented by some function $\Delta(|a_1 - a_2|)$. Using this representation, we consider the error of each interval $\left[\vec{a}_j^l, g_j^l\right]$:

$$\int_{\vec{a}_j^l}^{g_j^l} \frac{1}{\beta_j - \alpha_j} \Delta\left(\left|\vec{a}_j^l - t\right|\right) \mathrm{d}t$$

$$= \frac{1}{\beta_j - \alpha_j} \int_0^{g_j^l - \vec{a}_j^l} \Delta(t)\, \mathrm{d}t \tag{4}$$

Using exactly the same approach, we can also get the error of the interval $\left[g_j^l, \vec{a}_j^{l+1}\right]$:

$$\frac{1}{\beta_j - \alpha_j} \int_0^{\vec{a}_j^{l+1} - g_j^l} \Delta(t)\, \mathrm{d}t \tag{5}$$

We see that for any subgradient x of (4), $-x$ is a subgradient of (5). Using additivity of subdifferentials (see Rockafellar [33] chapter 23), we get that 0 is a subgradient of g_j^l. We can apply exactly the same approach to each $\vec{a}_j^l$ to see that 0 is a subgradient there as well. □

When the conditions of the above theorem do not hold, the decomposition results from Section 5.2 and the recursive linearization (3) still apply. In the following two subsections, we leverage this fact.

5.3.2 Convex error functions

As we pointed out above, taking the maximum, sum, and integral all preserve convexity. Thus, if the error

$$\left|\mu_z(a) - \mu_z(a')\right| \quad \text{or} \quad \left|\rho_s(a) - \rho(a')\right|$$

at each leaf or nature node can be represented by a convex function, the linear constraints in (3) can be used to represent the overall error as a convex function. As discussed in Section 5.1.1, this would, depending on the specific structure, allow the application of various efficient polynomial-time methods. We do not give specific algorithms here, but merely point out that optimal solutions can be found in polynomial time. In practice, the specific choice of which polynomial-time algorithm to apply should be informed by the exact structure of the error functions of the given game.

5.3.3 Piecewise linear error functions

We now consider piecewise linear utility error functions and piecewise linear nature probability error functions for discretizing continuous player action intervals. We do not consider discretizing nature actions here because even with linear functions and a uniform nature distribution, discretizing nature intervals would lead to quadratic error, as shown in the proof of Theorem 4. Even if the actual error functions are not piecewise linear, this can be used for arbitrarily accurate approximation.

Finding a bound-minimizing discretization, subject to a limit on the number of discretization points, is NP-hard. This is easily seen by considering the NP-hardness proof by Kroer and Sandholm [23], and representing their discrete game-abstraction problem using a step function.

However, it can be represented by a MIP, where the number of binary variables is equal to the number of pieces summed over all functions. This number can be significantly decreased if the interval pieces over the different functions μ, ρ under some interval $j \in \mathbb{I}$ align. We use the same variable formulation $\vec{a}_j^l, g_j^l$ as defined in (1), and the feasible set $\mathcal{F}$ remains the same. Consider an interval $j \in \mathbb{I}$ and the set of points where some function in the subtrees at j changes piece. This set of points divides the interval $[\alpha_j, \beta_j]$ into pieces. Let $\mathbb{P}_j$ be an index set into these pieces. The size of $\mathbb{P}_j$ is clearly bounded by the sum of pieces in functions in subtrees at j. We introduce a Boolean variable $b_j^{\gamma,l}, c_j^{\gamma,\tau}$ for each $\gamma \in \mathbb{P}_j$, $l \in [k_j]$, and $\tau \in [k_j - 1]$, representing whether $\vec{a}_j^l, g_j^\tau$ fall into the interval representing piece $\gamma \in \mathbb{P}_j$, respectively. When $b_j^{\gamma,l}(c_j^{\gamma,\tau}) = 1$, we restrict $\vec{a}_j^l(g_j^\tau)$ as follows:

$$\alpha_j^\gamma \cdot b_j^{\gamma,l} \leq \vec{a}_j^l, \quad \vec{a}_j^l \leq \beta_j^\gamma + (\beta_j - \beta_j^\gamma) \cdot b_j^{\gamma,l}$$

The sum $\sum_{\gamma \in \mathbb{P}_j} b_j^{\gamma,l} = 1$ ensures that only one interval is chosen. The constraints for $c_j^{\gamma,\tau}, g_j^\tau$ are completely analogous. For each leaf node z with piecewise payoff function μ_z^γ, and $l \in [k_j]$, we can then introduce price variables $p_z^l, p_z^l(\vec{a}_j^l), p_z^l(g_j^l) \in \mathbb{R}$, with the latter two for the interval $\left[\vec{a}_j^l, g_j^l\right]$, and constrain them linearly as follows:

$$\mathcal{P} = \left\{ \begin{array}{rll} p_z^l \geq & p_z^l(g_j^l) - p_z^l(\vec{a}_j^l) & \\ p_z^l(g_j^l) \geq & \mu_z^\gamma(g_j^l) - M \cdot c_j^{\gamma,\tau} & \forall \gamma \in \mathbb{P}_j \\ p_z^l(\vec{a}_j^l) \leq & \mu_z^\gamma(\vec{a}_j^l) + M \cdot b_j^{\gamma,l} & \forall \gamma \in \mathbb{P}_j \end{array} \right\} \tag{6}$$

This is correct for sufficiently large $M \in \mathbb{R}$. We now have variables p_z^l representing each error function for the discrete points, and can apply the linear constraints from (3) to get a linear representation of the overall error. Thus we get the following MIP, where e_r is the objective value at the root according to (3):

$$\min \left\{ e_\mathrm{r} : e_\mathrm{r} \in \mathcal{E}, \left(\mathcal{A}'_v, g_v\right) \in \mathcal{F} \cap \mathcal{P}, b_j^{\gamma,l}, c_j^{\gamma,\tau} \in \{0,1\} \right\} \tag{7}$$

6. APPLICATIONS

In this section, we discuss some applications of our results. We will focus on three recent problems that have included continuity in their problem formulation, or discretized away continuity: robust policy optimization under parameter uncertainty [8], security games [41, 22, 29], and sequential wifi-jamming under battery constraints [9].

Chen and Bowling [8] propose the use of zero-sum extensive-form game solving as a way of tractably computing optimally robust policies for Markov Decision Processes (MDPs) with parameter uncertainty. They design robustness criteria that can be implemented via nature first sampling parameter instances, and an opponent then choosing the worst of these. This sampling by nature was necessary in order to get games where the action space for the players is finite. Now, with our discretization-quality results, it is possible to use a

broader class of robustness measures that allow continuous action spaces, while obtaining solution quality bounds.

Several papers have investigated continuous settings for security games. These have been for single-shot [22, 41] or repeated Bayesian Stackelberg games [29]. Since our framework is for the more general setting of extensive-form games, our solution-quality bounds apply to all these settings. Furthermore, they would also apply to more sophisticated models that include both sequential actions and imperfect information. Marecki et al. [29] mention as future work the setting where the follower also behaves strategically. Our results immediately yield solution quality bounds for discretizations for this setting.

Another area with continuity is wifi jamming. In recent work, sequential-interaction models were introduced for this domain [9]. These models employ discretized action spaces, where both the jammer and transmitter have a (small) finite set of possible power levels to transmit at. However, this is an abstraction of reality, where software-defined radios mean that action spaces can be continuous (at least up to the bit-precision of the hardware). Using the techniques developed in this paper, we can give solution quality bounds on the utility loss obtained from considering only a discrete number of possible power levels (possibly by padding the game tree with dummy actions to satisfy Definition 1). DeBruhl et al. [9] also mention that in a more realistic model, both transmitter and jammer would be modeled as observing only noisy signals of the actions taken by the other player. Since these observations would be of a continuum, the noise would likewise be continuous. The discretization quality bounds derived here would immediately apply to this setting.

7. DIFFERENCES TO ABSTRACTION PRACTICE IN POKER

We have already discussed how our framework can be used to give theoretical bounds on solution quality in practical scenarios. In particular, we showed that a uniform discretization is optimal for linear error functions (for discretizing nature this required a uniform distribution over the continuous action space). This stands somewhat in contrast to how practical abstractions are created for poker.

Consider no-limit Texas hold'em (NLHE). This game is the premier testbed for (discrete) extensive-form game solving algorithms [34]. Each year, the Annual Computer Poker Competition is held, where research groups submit highly-tuned poker-playing programs. The winning programs are based on computing Nash equilibrium approximations in abstractions of the full extensive-form game [34].

In NLHE, at each betting step, the acting player may bet any amount from the minimum bet to their entire stack of chips. To handle this action space, the top agents devise betting abstractions. These are completely analogous to the discretizations considered in this paper. The payoff functions under the subtrees are all linear in the specific actions chosen. At a cursory glance, one might say that Theorem 4 suggests that the optimal discretization would be uniform. However, the discretizations employed by the poker bots are more akin to a geometric progression. For example, Brown et al. [7] describe using a betting abstraction consisting of $0.5, 0.75, 1, 1.5, 2$ and 5 times the pot size, as well as the all-in action. At the start of betting, all-in is approximately 133 times the pot size. Both examples and experiments by Ganzfried and Sandholm [10] support the idea that the uniform mapping is not optimal.

A potential explanation for this is that the subtrees reached for different choices of bet size technically do not fall under the constraints of Definition 1. Consider a group of bets, say raising in the range of $[1, 2]$ (here we consider continuous bets in this small continuous range due to ease of exposition, one can construct similar examples with larger integer ranges) with a stack size of 2.5. The subtree where the player bets 2 exists as a subset of the subtree at every other betsize. These subsets all map to each other in a way that obeys Definition 1. However, if the player bets 1 instead, the opponent may reraise by 1, in which case the agent can call. This scenario does not exist when betting 2, as the player already bet her entire stack. For any two bet sizes $a_1 < a_2$, these discrepancies due to extra actions exist. To resolve this issue, one could pad the tree with extra actions such that Definition 1 is satisfied, but it is unclear how this would interact with the solution-quality bound, and could thus lead to nonoptimality of the uniform discretization.

8. CONCLUSIONS

We analyzed the problem of developing solution-quality bounds for discretization of extensive-form games with continuous action spaces. To our knowledge, we developed the first such bounds. We developed bounds for a very general class of continuous games: ones where there is a finite set of prototype trees, such that the instantiation of a given prototype has its nature probability outcomes and utilities parameterized by the choice on the continuous intervals. We developed bounds both where they depend on the specific Nash equilibrium (Theorem 2) and where they are independent of the Nash equilibrium chosen (Corollary 3).

We then considered the problem of computing bound-minimizing discretizations. First we developed very general bound-minimization formulations that allow a broad class of error-bounding functions. We discussed how such functions can be minimized in polynomial time when they are convex. We then considered the more specific problem of minimizing our bound developed for Corollary 3. For the case where all utility error and nature probability error functions are Lipschitz continuous (without additional structure), and nature chooses uniformly over each continuous interval, we showed that the bound-minimizing solution is to discretize uniformly. For the case where the error functions at individual nodes can be represented by convex functions, we showed how to generate a convex optimization formulation of the overall problem. We also developed a MIP for piecewise linear error functions, which can also be used for arbitrarily accurate approximation. We also showed how the problem can be decomposed into separately optimizable components.

Ganzfried and Sandholm [10] dicuss randomized mappings, where each real point has a probability distribution over which of its nearest discrete points it maps to. Their experiments strongly suggest that such randomization is desirable. Incorporating randomized mappings in our work could potentially lead to better discretizations, almost certainly in practice, and potentially also in theory. We leave this as future research.

Acknowledgements. This material is based on work supported by the National Science Foundation under grant IIS-1320620.

References

[1] C. Archibald and Y. Shoham. "Modeling Billiards Games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Budapest, Hungary, 2009.

[2] Nicola Basilico and Nicola Gatti. "Automated Abstractions for Patrolling Security Games". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2011.

[3] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Vol. 2. Siam, 2001.

[4] Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. "Approximating Game-Theoretic Optimal Strategies for Full-scale Poker". In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. 2003.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] Noam Brown and Tuomas Sandholm. "Regret Transfer and Parameter Optimization". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2014.

[7] Noam Brown, Sam Ganzfried, and Tuomas Sandholm. "Hierarchical Abstraction, Distributed Equilibrium Computation, and Post-Processing, with Application to a Champion No-Limit Texas Hold 'em Agent". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2015.

[8] Katherine Chen and Michael Bowling. "Tractable Objectives for Robust Policy Optimization". In: *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2012.

[9] Bruce DeBruhl, Christian Kroer, Anupam Datta, Tuomas Sandholm, and Patrick Tague. "Power napping with loud neighbors: optimal energy-constrained jamming and anti-jamming". In: *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*. ACM. 2014, pp. 117–128.

[10] Sam Ganzfried and Tuomas Sandholm. "Action Translation in Extensive-Form Games with Large Action Spaces: Axioms, Paradoxes, and the Pseudo-Harmonic Mapping". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2013.

[11] Andrew Gilpin and Tuomas Sandholm. "A Competitive Texas Hold'em Poker Player via Automated Abstraction and Real-Time Equilibrium Computation". In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 2006, pp. 1007–1013.

[12] Andrew Gilpin and Tuomas Sandholm. "Lossless Abstraction of Imperfect Information Games". In: *Journal of the ACM* 54.5 (2007). Early version in EC-06.

[13] Andrew Gilpin and Tuomas Sandholm. "Expectation-Based versus Potential-Aware Automated Abstraction in Imperfect Information games: An Experimental Comparison Using Poker". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Short paper. 2008.

[14] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. "A Heads-Up No-Limit Texas Hold'em Poker Player: Discretized Betting Models and Automatically Generated Equilibrium-Finding Programs". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2008.

[15] John Hawkin, Robert Holte, and Duane Szafron. "Automated action abstraction of imperfect information extensive-form games". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2011.

[16] John Hawkin, Robert Holte, and Duane Szafron. "Using sliding windows to generate action abstractions in extensive-form games". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2012.

[17] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. "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.

[18] Albert Jiang and Kevin Leyton-Brown. "Polynomial-time computation of exact correlated equilibrium in compact games". In: *Proceedings of the ACM Conference on Electronic Commerce (EC)*. 2011.

[19] Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. "Accelerating Best Response Calculation in Large Extensive Games". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2011.

[20] Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. "Finding Optimal Abstract Strategies in Extensive-Form Games". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2012.

[21] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. "Evaluating State-Space Abstractions in Extensive-Form Games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2013.

[22] Christopher Kiekintveld, Janusz Marecki, and Milind Tambe. "Approximation methods for infinite Bayesian Stackelberg games: modeling distributional payoff uncertainty". In: *10th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2011*. 2011.

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

[24] Christian Kroer and Tuomas Sandholm. *Extensive-Form Game Imperfect-Recall Abstractions With Bounds*. arXiv. 2014.

[25] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. "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.

[26] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. "No-Regret Learning in Extensive-Form Games with Imperfect Recall". In: *International Conference on Machine Learning (ICML)*. 2012.

[27] Richard Lipton, Evangelos Markakis, and Aranyak Mehta. "Playing Large Games Using Simple Strategies". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. 2003.

[28] Michael Littman and Peter Stone. "A polynomial-time Nash equilibrium algorithm for repeated games". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. 2003.

[29] Janusz Marecki, Gerald Tesauro, and Richard Segal. "Playing repeated Stackelberg games with unknown opponents". In: *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012*. 2012.

[30] Yurii Nesterov. "Smooth Minimization of Non-Smooth Functions". In: *Mathematical Programming*. A 103 (2005), pp. 127–152.

[31] Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[32] FranÃğois Pays. "An Interior Point Approach to Large Games of Incomplete Information". In: *AAAI Computer Poker Workshop*. 2014.

[33] R. Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[34] 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.

[35] 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, Senior Member Track*. 2015.

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

[37] David Schnizlein, Michael Bowling, and Duane Szafron. "Probabilistic State Translation in Extensive Games with Large Action Sets". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 2009.

[38] Jiefu Shi and Michael Littman. "Abstraction Methods for Game Theoretic Poker". In: *CG '00: Revised Papers from the Second International Conference on Computers and Games*. 2002.

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

[40] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. "Abstraction Pathologies in Extensive Games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2009.

[41] Zhengyu Yin and Milind Tambe. "A unified method for handling discrete and continuous uncertainty in Bayesian Stackelberg games". In: *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012*. 2012.

[42] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. "Regret Minimization in Games with Incomplete Information". In: *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2007.